The 27th annual workshop of the Swedish Artificial Intelligence Society (SAIS)

14–15 May 2012, Örebro, Sweden

Editors

Lars Karlsson

Julien Bidot

Copyright

Linköping Electronic Conference Proceedings, No. 71
Linköping University Electronic Press
Linköping, Sweden, 2012
ISSN: 1650-3686
eISSN: 1650-3740
URL: http://www.ep.liu.se/ecp_home/index.en.aspx?issue=071

Conference Organization

Programme Chairs

Lars Karlsson
Julien Bidot

Programme Committee

Marcus Bjäreland
Eva Blomqvist
Henrik Boström
Paul Davidsson
Patrick Doherty
Göran Falkman
Robert Feldt
Peter Funk
Fredrik Heintz
Anders Holst
Magnus Johnsson
Danica Kragic
Niklas Lavesson
Helena Lindgren
Jacek Malec
Lars Mollberg
Cecilia Sönströd

Local organization

Barbro Alvin
Mia Lindström
Lia Susana D.C. Silva-Lopez

Table of Contents

The 27th annual workshop of the Swedish Artificial Intelligence Society (SAIS),
14–15 May 2012, Örebro, Sweden
Published by Linköping University Electronic Press
http://www.ep.liu.se/ecp_home/index.en.aspx?issue=071

Ideas for Fault Detection Using Relation Discovery

Sławomir Nowaczyk
Halmstad University
Slawomir.Nowaczyk@hh.se

Rune Prytz
Volvo Group Trucks Technology
Rune.Prytz@volvo.com

Stefan Byttner
Halmstad University
Stefan.Byttner@hh.se

Abstract

Predictive maintenance is becoming more and more important in many industries, especially taking into account the increasing focus on offering uptime guarantees to the customers. However, in automotive industry, there is a limitation on the engineering effort and sensor capabilities available for that purpose. Luckily, it has recently become feasible to analyse large amounts of data on-board vehicles in a timely manner. This allows approaches based on data mining and pattern recognition techniques to augment existing, hand crafted algorithms.

Automated deviation detection offers both broader applicability, by virtue of detecting unexpected faults and cross-analysing data from different subsystems, as well as higher sensitivity, due to its ability to take into account specifics of a selected, small set of vehicles used in a particular way under similar conditions.

In a project called Redi2Service we work towards developing methods for autonomous and unsupervised relationship discovery, algorithms for detecting deviations within those relationships (both considering different moments in time, and different vehicles in a fleet), as well as ways to correlate those deviations to known and unknown faults. In this paper we present the type of data we are working with, justify why we believe relationships between signals are a good knowledge representation, and show results of early experiments where supervised learning was used to evaluate discovered relations.

1 Introduction

Most industries nowadays are moving towards more sophisticated cyber-physical systems, with new challenges arising from increased software and system complexity. Developing and, even more importantly, maintaining those systems requires a significant engineering effort. For commercial ground vehicle operators (such as bus and truck fleet owners), the maintenance strategy is typically reactive, meaning that a fault is fixed only after it has become an issue affecting vehicle's performance. Uptime guarantees consist in scheduling component maintenance and replacement based on statistical lifetime predictions.

The biggest difficulty in moving towards predictive maintenance, in the vehicle industry, lies in limited budget for on-board sensors and the amount of engineering time it takes to develop diagnostic algorithms. Predicting that there is a need for maintenance *before* something breaks down is virtually impossible to plan during vehicle development cycle, especially if diagnostic algorithms need to handle multiple different kinds of faults, work in a consistent manner on a wide variety of vehicle configurations, as well as for many different types of operation under varying environment conditions.

The development costs of fault diagnostics in the classical paradigm will keep growing, with the current trend of increasing number of components in vehicles and stricter requirements on their efficiency. The only solution seems to be augmenting engineering work with automated data analysis. This has been made possible by the introduction of low-cost wireless communication. Data mining can now be performed on-board real vehicles as they are being used. The subsystems that are critical for safety or long-term health of the vehicle will always use, to some degree at least, diagnostic mechanism developed and tested by engineers, but there is a lot of value to be gained from monitoring the state of as many additional subsystems as possible.

Our approach is based on unsupervised discovery of relations between various signals that are available on the internal vehicle network. While it is difficult to detect faults by looking at characteristics of signals (such as *road speed*) in isolation, the interrelations of connected signals are more indicative of abnormal conditions.

The difference between our work and most other approaches lies in the requirement that relation discovery be done completely autonomously. While engineers are often able to propose a large number of "good" relations between various signals, those are typically general enough to hold for all or almost all vehicles. A data mining system we aim to develop, however, should be tightly connected to a particular *fleet* of vehicles, either by geographic region, vehicle configurations or type of operation. Some of the relations that are useful for detecting faults in long-haul trucks would be inadequate for delivery trucks, for example. Today, it is not feasible to develop specialised diagnostic algorithms for each of those cases, even though they would be very useful.

Therefore, the two main benefits of automated deviation detection are broader applicability and higher sensitivity. The former is obtained by no longer requiring an expert to target a particular subsystem, analyse it in isolation, predict possible faults, what their symptoms would be, and which of those symptoms can be guaranteed to never appear during normal operation. Instead, the data mining approach can take a more "complete vehicle" perspective, both cross-checking data from different subsystems, as well as detecting faults nobody has predicted can take place. There are many potential problems that are currently not being monitored, since the cost of hand-crafting diagnosis methods for them is too high. We expect fully automatic methods to be less reliable than engineered and heavily tested diagnostic routines, but they are not intended to replace, but rather supplement them.

Higher sensitivity can be achieved because unsupervised deviation detection algorithms can take a fleet-based approach, where "normal operation" can be defined on a much smaller scale. There are very few subsystems in a vehicle where manufacturers can afford to develop a diagnostic method that is specialised (or at least parameterised) for a particular geographical region or a particular usage pattern. In the case of pattern recognition approaches, it is very easy to define "standards" for a particular location or for a particular type of operation — or even for both at the same time, for example by only comparing vehicles within a single company. This can lead to earlier detection of problems, where a fleet owner is alerted that a given symptom is unusual *for their vehicles*, even though the same symptom would be perfectly normal for another operator. This is becoming more important as new hardware and, especially, software capabilities in vehicles lead to higher customisability of the manufacturers' offer and mean that "one size fits all" solutions are becoming less desirable.

This paper is organised as follows. We briefly present related research in the next section, followed by description of data we are working with in Section 3. We discuss signal relations in Section 4 and ways to discover them in 5. We close by evaluation in Section 6 and conclusions in Section 7.

2 Related research

Automated data mining for vehicle applications has previously been the topic of several papers. An early paper by Kargupta et. al. [4] shows a system architecture for distributed data-mining in vehicles, and discusses the challenges in automating vehicle data analysis. In Zhang et al. [9], being able to do cross-fleet analysis (comparing properties of different vehicles) is shown to benefit root-cause analysis for pre-production diagnostics. In Byttner et. al. [1], a method called COSMO is proposed for distributed search of "interesting relations" (e.g. strong linear correlations that hold for long periods of time) among on-board signals in a fleet of vehicles. The interesting relations can then be monitored over time to enable e.g. deviation detection in specific components. A method based on a similar concept of monitoring correlations (but for a single vehicle instead of a fleet) is shown in D'Silva [2]. In Vachkov [8], the neural gas algorithm is used to model interesting relations for diagnostic of hydraulic excavators. Contrary to our work, however, both the papers by D'Silva and Vachkov assume that the signals which contain the interesting relations are known *apriori*. In [5], a method for monitoring relations between signals in aircraft engines is presented. Relations are compared across a fleet of planes and flights. Unlike us, however, they focus on discovering relationships that are later evaluated by domain experts.

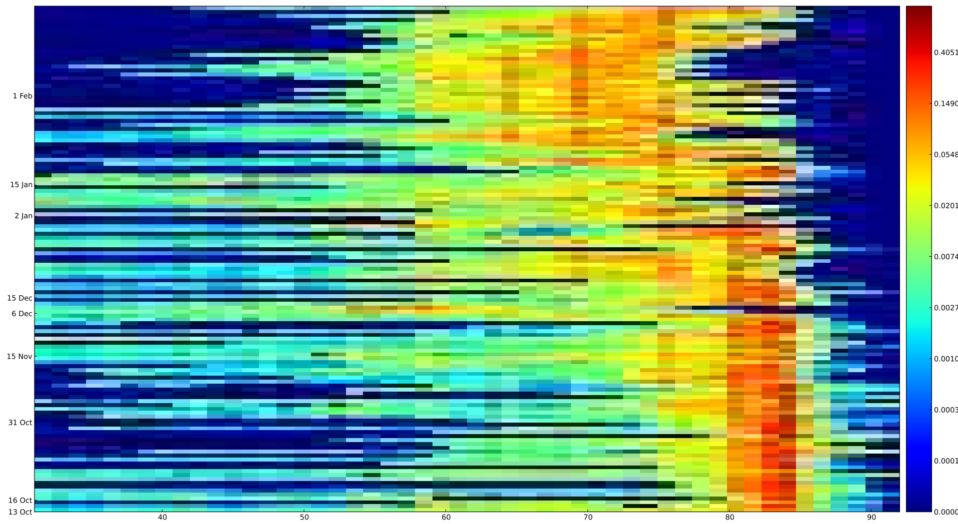

Figure 1: Engine Coolant Temperature

3 Description of data

As a part of the Redi2Service project, we have equipped 19 Volvo buses with the hardware capable of collecting data from the internal vehicle network. Our setup has been in place since September 2011, giving us almost half a year worth of data from their operation in western Sweden. The data consists of over 100 signals that are measured with a sampling frequency of 1 Hz. This results in a volume of roughly 10GB of data per week. As an example, in Figure 2 we plot the values of a signal called *Vehicle Speed* over a half an hour episode.

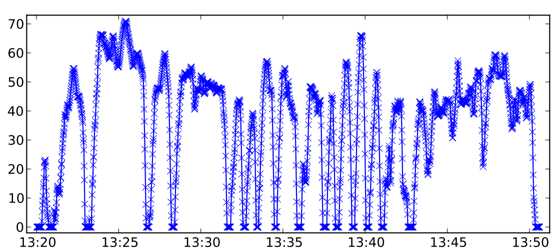

Figure 2: Vehicle Speed

This gives us access to data from a real-world operation of those vehicles, without any artificial modification to either the usage or condition of the buses. On one hand, this makes the data extremely valuable, but at the same time makes it difficult to analyse, since we cannot assume that all the buses are in the perfect initial condition. In fact, we are already seeing symptoms that indicate significant number of technical problems in some of them.

The most natural way of analysing such data is by using histograms. This is not the only way, of course, and there are a number of characteristics that are not captured by a histogram, but this is a good starting point. In the abstract, before bringing human expert knowledge to the table, we are interested in differences in signal characteristics along two axes: between different time periods and between different vehicles.

In this paper we will focus on the first one. We need to analyse how does a particular signal change in time, hopefully leading to a discovery of components that are starting in good condition, slowly wear out, until they reach a point where they can be considered "broken" and they start to negatively affect the performance of the whole bus.

A promising trend can be seen in Figure 1, where we plot a sequence of histograms for the signal called *Engine Coolant Temperature* over a period of four months. Each horizontal line corresponds to a set of 20000 data readings, presented as a colour map histogram with logarithmic scale. The bar on the right visualises how the value probabilities are mapped into colours. It is interesting to note that the actual amount of data we obtain from each bus varies significantly in time, according to usage in a given period. Therefore, while the Y axis in Figure 1 represents real time, it is definitely not linear, as indicated by the dates shown.

This plot, however, reveals a critical flaw in looking at signals in isolation. To a human expert Figure 1 does not indicate a trend corresponding to a wearing out component, but rather an influence of a well-known external condition: it is simply significantly colder in January than it is in October.

4 Signal Relations

One possible way to increase robustness against external influences is to look at relations between different but related signals. Our claim — supported, to some degree at least, both by the data we have been collecting now and by results of earlier experiments — is that there exist a large number of "interesting" relationships between signals and that those are a better predictors of faults than characteristics of individual signals alone.

One such example would be relation between signals *Oil Temperature* and the aforementioned *Engine Coolant Temperature*, depicted in Figure 3. It is more difficult to visualise such relation across multiple time periods, and so we have decided to only present a scatter plot for October 2011 and for January 2012, each containing 40000 readings.

As can be expected due to the basic laws of thermodynamics, there is a strong linear relation between those two signals. The plots are definitely not identical (for example, both signals reach higher values in October), but there is a fundamental structure to the relation that has not changed. Our goal is to capture this in a model. Faults that affect one of the subsystems but not the other would then introduce a systematic shift that would change parameters of that model.

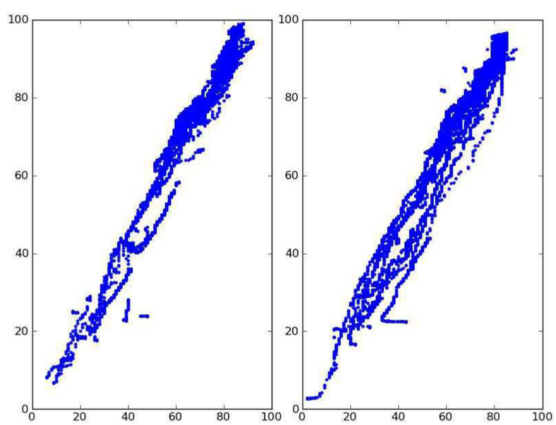

Figure 3: Scatter plot of Oil Temperature against Engine Coolant Temperature, October and January

Of course, relations between signals can be arbitrarily complex, but one aspect of the hardware installed in our buses is it's capability to perform pattern recognition on-board. Right now we are storing all the data on USB memories and only mining this data off-line, in order to better understand what we are dealing with. Ultimately, however, the model estimation will be done on the vehicle and only the result of it will be transmitted, wirelessly, to a central server. There, they will be compared with data from the past and from other members of the fleet, and decision will be made whether any deviations are interesting enough to show the user.

Due to limited computational power, we have mostly limited ourselves to looking at linear models. We are investigating other solutions, however, since non-linear relations are quite common.

An important resource is also a database called *Vehicle Service Records*, which contains a detailed information about every repair and maintenance operation during the lifetime of a bus. It will allow us to not only inform the user that there is a problem with their vehicle, but also what had to be done to fix it last time similar thing happened.

5 Relation discovery

In general, it is far from trivial to evaluate ideas we have presented in the previous sections. The only true measure is the savings in maintenance expenses once the system is deployed. However, as a start, we have performed an experiment on a Volvo VN780 truck, where we have been collecting values of 21 signals, over 10 driving runs with four different faults injected, as well as 4 runs under normal operating conditions. Each episode lasted approximately four hours, and took place in a controlled environment under a variety of driving situations. The exact details of faults are not important here, but they include clogged of *Air Filter* and *Grill*, leaking *Charge Air Cooler* and partially congested *Exhaust Pipe*.

The method we used for discovering relations consists of three steps. We start with data preprocessing and removing the influence of ambient conditions, but we do not discuss details here, interested readers can find them in [6]. We then proceed to choose the most interesting signals to model, as well as which signals should be used to model them. Finally, we estimate model parameters.

The main challenge is to determine which relations exist between signals. We begin by modelling each signal using all other signals as regressors:

$$\Psi_k = \underset{\Psi \in \mathbb{R}^{s-1}}{\arg\min} \left(\sum_{t=1}^{n} \left(y_k(t) - \Psi^\top \varphi_k(t) \right)^2 \right) \qquad (1)$$

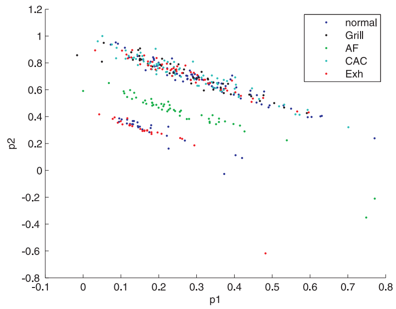

Figure 4: Model parameters (Lasso method)[1]

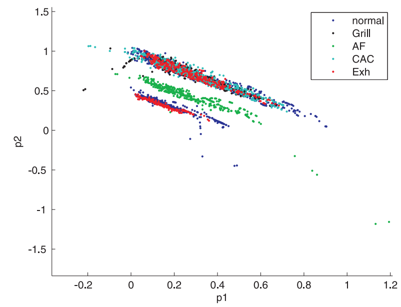

Figure 5: Model parameters (RLS method)[1]

where s is number of signals, Ψ_k is a vector of parameter estimates for the model of y_k and φ_k is the regressor for y_k (i.e. the set of all *other* signals).

Following the LASSO (Least Absolute Shrinkage and Selection Operator) method [7], we use an energy constraint C_k as an upper bound on the sum of absolute values of all parameters for y_k:

$$\sum_{i=0}^{s-1} ||\Psi_{k,i}|| < C_k \tag{2}$$

We gradually increase value of C_k, performing a cross-validation test after each run. Initially, the mean squared error of the model keeps decreasing, but at some point it begins to increase, as it starts to overfit. We then make a decision of whether the MSE is sufficiently low to consider this model to be good enough. Different C_k are optimal for each model, with some never reaching below the chosen MSE threshold and thus being considered uninteresting for further analysis.

The second stage consists of finding and removing insignificant model parameters, namely those which are unstable and with low values. To this end, a sequence of estimates for each regressor within a model is collected over a series of time slices. We perform a *t-test* to find which of those estimates are significant, i.e. which are non-zero. This allows us to remove artificial signal dependencies, leaving only strong relationships.

For calculating parameters for the selected models at difference times, we have tested two different approaches. The first is the LASSO method as outlined above, where we split data into a number of time slices, and, for each slice, calculate optimal model parameters. The second method is RLS (Recursive Least Squares) method [3], which recursively calculates the estimates over a sliding window defined by the forgetting factor:

[1]previously published in [6]

$$P(0) = \delta_{init}^{-1} I \qquad\qquad \Theta(0) = \Theta_{init} \tag{3}$$

$$e(n) = y(n) - \Theta^\top(n-1)\varphi(n) \tag{4}$$

$$g(n) = \frac{P(n-1)\varphi(n)}{\lambda + \varphi^\top(n)P(n-1)\varphi(n)} \tag{5}$$

$$P(n) = \lambda^{-1}P(n-1) - g(n)\varphi^\top(n)\lambda^{-1}P(n-1) \tag{6}$$

$$\Theta(n) = \Theta(n-1) + e(n)g(n) \tag{7}$$

The reason we chose those two methods is that LASSO approach allows an estimator to easily adapt to models that are changing in time, at the cost of possible oscillating behaviour if several models are of similar quality. On the other hand, RLS offers very fast convergence, but — due to it's incremental nature — takes a long time to "catch up" if the underlying relation changes.

As an example, Figures 4 and 5 show parameters in "fuel inst $= p_1 *$ cac in p $+ p_2 *$ in manif t" relation: *fuel inst* (instantaneous fuel consumption) can be approximated using *cac in p* (charge air cooler input pressure) and *in manif t* (input manifold temperature). We plotted estimates obtained by both LASSO and RLS methods using colours to mark which fault was injected during a particular run.

From among our four faults, only clogged *Air Filter* (AF) can be discovered based on the *fuel inst* relation above. There are other relations that are useful for other faults, of course, but it is difficult to get a clear overview of the complete solution.

6 Evaluation

As shown in the previous section, some faults can be detected reasonable easily. Unfortunately, it does not hold for all of them. Actually, the biggest problem with the experiment as we have done it is that some of the injected faults were easy, while others were very difficult to detect.

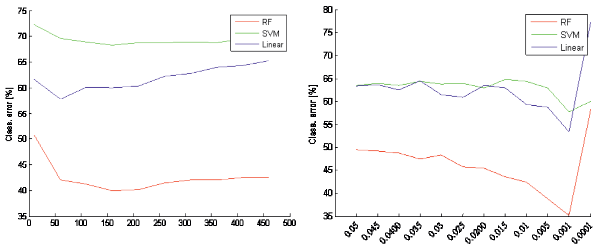

Figure 6: Classification error, Lasso and RLS[1]

We have decided to use supervised learning to evaluate whether parameters of our models (both from LASSO and RLS estimators) can be useful for detecting faults. We tried three different classifiers: linear regression, support vector machine (SVM) and random forest. Both the forgetting factor (for RLS) and the number of data slices (for LASSO) are parameters for tuning. We have found that less slices and larger forgetting factor gives better signal to noise ratio and a more robust solution. However, they are a lot less sensitive to faults that are only apparent under certain conditions. This is due to the smoothing larger slices and forgetting factor results in. As an example, a partially clogged air filter will only have a visible effect if the engine is running at high power, since this is the only situation when a large air flow is required.

We have run the classification task a number of times, varying the time slice size the and forgetting factor. It is easy to see from Figure 6 that choosing too small forgetting factor for RLS is detrimental. On the other hand, the effect of choosing too many data slices is hardly visible.

In general, the random forest classifier outperforms both SVM and linear classifier by a pretty large margin. We do not know why this is the case, since we have not investigated the classification itself in great depth. More interestingly, RLS estimator appears to give slightly better results than the LASSO estimator, but it probably is not worth the increased computational complexity.

As a final comment, the resulting classification error appears to be rather high, but it is important to take into account that this data set is a very difficult one. There is a lot of different external influences that disturb the "normal" operation of a truck, and the low quality of available sensors result in high levels of noise in the data. The lack of dedicated sensors is also a problem: neither of the four faults we have analysed is being monitored in any way for current in-production vehicles.

7 Conclusions

In this paper we present a project that we are involved in, developing an unsupervised algorithm for discovering interesting relations between time series of vehicle signal data, to be used for fault detection and predictive maintenance. We present our approach and show initial evaluation, using supervised learning, on the data collected from a Volvo truck during a fault injection experiment.

This is a step towards a system that would be able to analyse on-board data on real vehicles and detect anomalies in an autonomous way. Ideas presented here are very much work in progress and there are numerous directions to extend those results. Primarily, we have not really answered the question of how to distinguish "interesting" relations from "uninteresting" ones, especially taking into account that we are looking for those that hold *most*, but definitely not *all*, of the time.

It is also not quite clear if the supervised classification is the best way of evaluating usefulness of discovered relations. We intend to explore other possibilities, especially those connected to the service records database we have access to.

Acknowledgements. This work has been partially funded by grants from VINNOVA and from the Knowledge Foundation.

References

[1] S. Byttner, T. Rögnvaldsson, and M. Svensson. Consensus self-organized models for fault detection (COSMO). *Engineering Applications of Artificial Intelligence*, 24(5):833–839, 2011.

[2] S.H. D'Silva. Diagnostics based on the statistical correlation of sensors. Technical Report 2008-01-0129, Society of Automotive Engineers (SAE), 2008.

[3] Monson H. Hayes. *Statistical Digital Signal Processing and Modeling*. John Wiley & Sons, Inc., 1996.

[4] H. Kargupta et al. VEDAS: A mobile and distributed data stream mining system for real-time vehicle monitoring. In *Int. SIAM Data Mining Conference*, 2003.

[5] J. Lacaille and E. Come. Visual mining and statistics for turbofan engine fleet. In *IEEE Aerospace Conf.*, 2011.

[6] Rune Prytz, Sławomir Nowaczyk, and Stefan Byttner. Towards relation discovery for diagnostics. In *Proceedings of the First International Workshop on Data Mining for Service and Maintenance*, pages 23–27, 2011.

[7] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58, 1996.

[8] G. Vachkov. Intelligent data analysis for performance evaluation and fault diagnosis in complex systems. In *IEEE International Conference on Fuzzy Systems*, pages 6322–6329, July 2006.

[9] Y. Zhang, G.W. Gantt, et al. Connected vehicle diagnostics and prognostics, concept, and initial practice. *IEEE Transactions on Reliability*, 58(2), 2009.

The 27th annual workshop of the Swedish Artificial Intelligence Society (SAIS),
14–15 May 2012, Örebro, Sweden
Published by Linköping University Electronic Press
http://www.ep.liu.se/ecp_home/index.en.aspx?issue=071

Personalisation and User Models for Support in Daily Living

Jayalakshmi Baskar, Helena Lindgren, Dipak Surie, Chunli Yan and Farahnaz Yekeh
Department of Computing Science
SE-901 87 Umeå
Sweden
{jaya,helena,dipak,chunli,yekeh}@cs.umu.se

ABSTRACT

In recent years, the interest in developing personalised applications for home environment has grown since it has a wide reach in helping people in their daily activities. However, for our purposes the concept activities of daily living also need to include work and leisure activities not necessarily performed in home environments. In this article, we describe an ongoing effort to develop a generic framework for assessing ability and tailoring of support applications in the health domain. We also give an overview of the approaches that have been adopted for personalisation and user modelling to various application areas. Suggestions of future development are provided.

KEYWORDS

Personalisation, User modelling, User models, Clinical decision-support systems, Multi-agent systems, Ambient intelligence, Human computer interaction, Adaptive hypermedia system, E-learning, Adaptive education systems, Machine learning, Intelligent tutoring systems

1. Introduction

Web-based applications have users who have different knowledge, learning styles, interests, background and preferences regarding information presentation over the Internet. This has paved way to research on interfaces that can be designed to recognize the goals and characteristics of the user and adapt accordingly. In order to achieve adaptability of personalised information, it is important to observe the user's behaviour, and make predictions based on those observations. The information pertaining to individual user obtained from such observations is known as a *user model* (e.g., [1]). A user model, or a simple user profile, may consist of information collected by filling questionnaires, by observing user-actions, or by making inferences. *Personalisation* aims at providing users with the content that they need without necessarily requiring the users to specify it explicitly [2].

Applications developed for smart homes or for the healthcare domain need to be adaptable to the needs of an individual. Therefore, customizing the environment services according to the user preferences is very important. The personalisation for dependent people is a difficult task, which should involve a team of disciplines such as ergonomics, occupational therapy, design, engineering, medicine etc. Furthermore, the development of tailored educational systems for knowledge workers such as medical or mining personnel also requires knowledge in the different knowledge domains.

In the work on developing knowledge-based applications tailored to individuals in three different knowledge domains, a common user model has been synthesized based on pilot projects [3]. The pilot projects have been targeting the diagnosis of dementia, monitoring health in the mining and construction industries and activity support for older adults in their home environments. The common user model is implemented as an RDF/OWL ontology functioning as part of ACKTUS, a knowledge and interaction modelling prototype application for the health domain. The knowledge and interaction is primarily modelled by domain experts. Consequently, focus has been on their explicitly defined user scenarios where they adapt the knowledge to different characteristics by using simple rules. However, there is a need to extend the tailored support by supplementing the adaptability of ACKTUS applications with adaptive functionality. There are three particular goals for this work: 1) to tailor support to an individual health professional's diagnostic reasoning; 2) to tailor support to an individual mining worker based on a combination of self-assessed complaints, body measures and factors obtained in his/her work environment; and 3) to tailor support for an individual in their home environment based on a combination of self-assessments and activity recognition and evaluation. In this work, focus is set on the third goal, however, examples are provided also for the first two. The purpose of the work presented in this paper is to present and evaluate to what extent the current user model applied in ACKTUS is sufficient for the purpose and in what aspects the model needs to be developed. For this purpose, existing approaches to the task are being explored and evaluated.

This paper is organised as follows. Section 2 describes ACKTUS, which is a prototype to be used by domain professionals for modelling personalised behaviour and user models. Section 3 describes personalisation research similar to the approach presented in this paper. In Section 4 the results are summarised and future work is outlined.

2. ACKTUS - a Tool for Modelling Personalised Behaviour and User Models

Empirical studies of three different knowledge domains generated features essential for the different knowledge domains in the personalisation of support applications [3]. These features were categorized and interpreted using activity theory [4] and existing medical and health terminologies. A model, which functions as a generic model common for the different domains, was developed as core ontology. The backbone of the ontology utilizes activity theoretical models for human in activity and the ACKTUS user model is partly built upon the International Classification of Functionality, Ability and Health (ICF) developed by WHO[1]. The core ontology is extended in the modelling of domain specific content by the domain experts. In this modelling, the domain experts can construct user models by combining features as RDF-tuples using dedicated user interfaces (Figures 1-6).

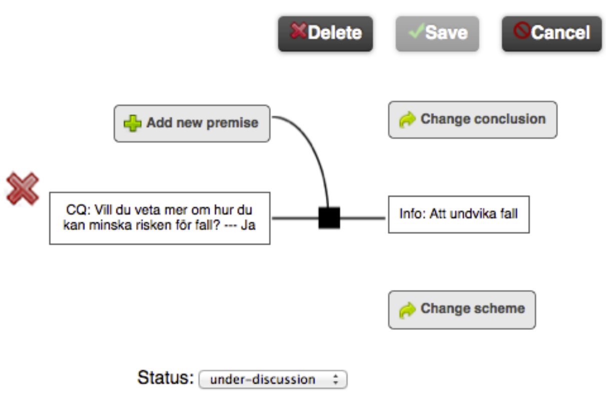

Figure 1. The editor for composing rules. An example of an advice to be given to an end user if interest is expressed.

When the user provides information in an end-user application, the application requests further information depending on which features are provided, following the thematic protocols of information collection defined by the domain experts. Based on this, the user profile is built.

In the system's dialogues with the user, the user is provided responses from the system in the form of 1) suggestions of decisions, 2) advices and 3) suggestions of actions to make. Currently, the actions are formally defined as ACKTUS assessment protocols.

Figure 2. The editor for composing rules to be used in assessments. A premise can be added in this example.

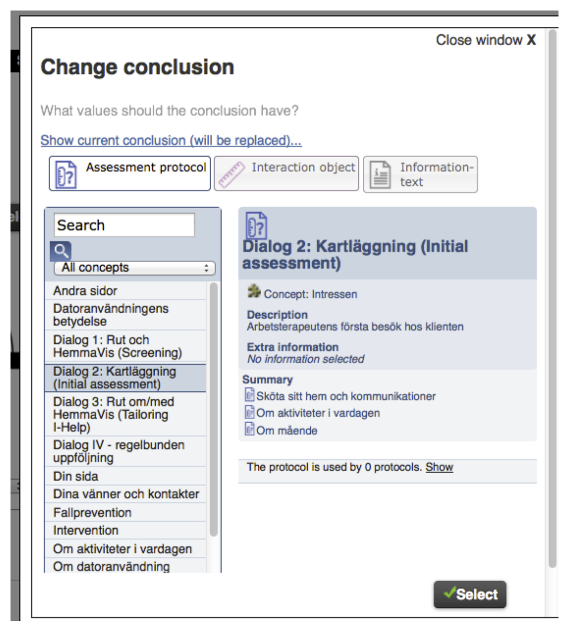

Figure 3. Example of an attempt to change the consequent of a rule to an ACKTUS assessment protocol.

[1] http://www.who.int/classifications/icf/en/

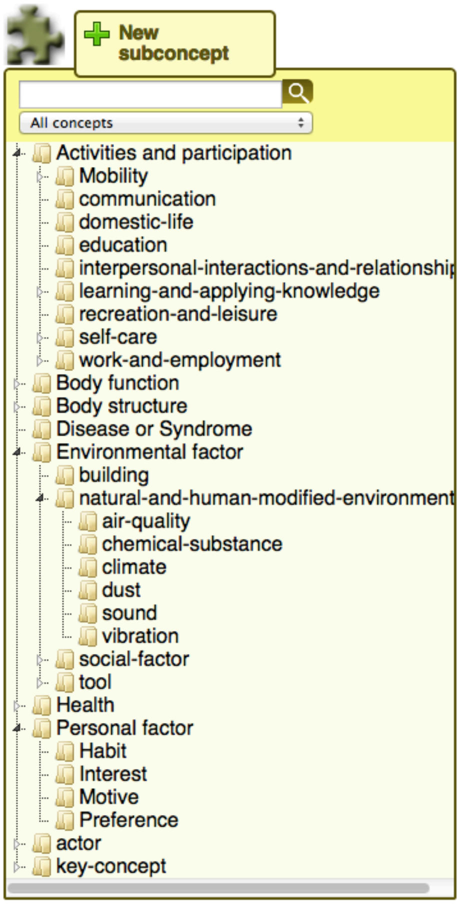

Figure 4. An overview of the nodes in the core ontology that builds the user models. The domain experts can refine the conceptual model by creating sub-classes so that it suits the domain.

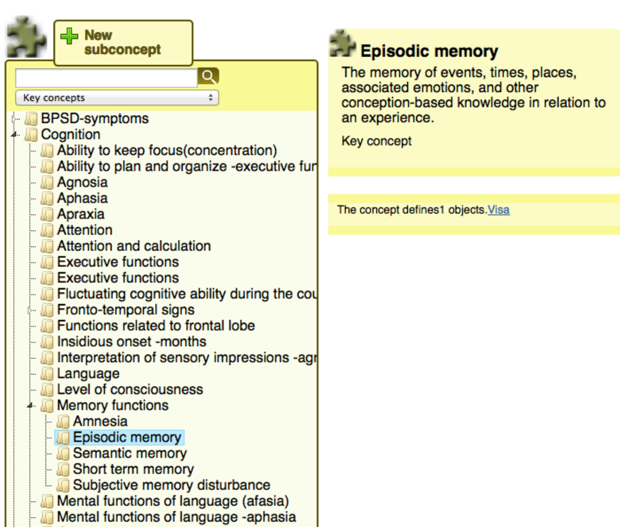

Figure 5. Example of a sub-tree of concepts particular to the dementia domain specified by domain experts. Definitions or explanations can be added, used for educational purposes in end user applications.

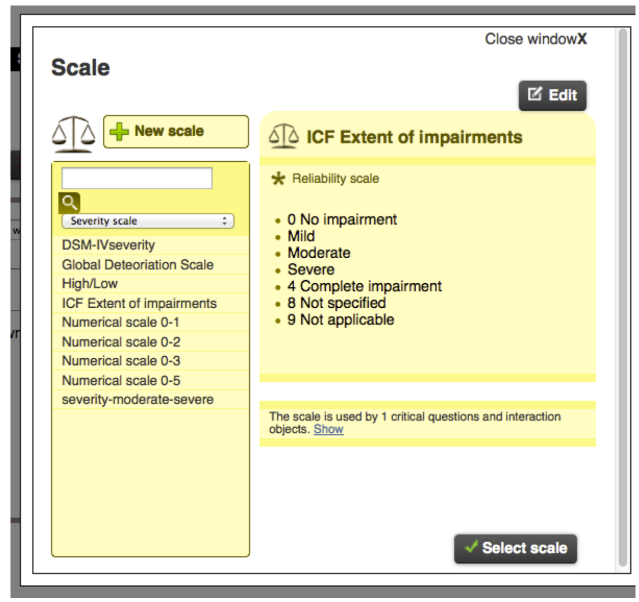

Figure 6. Scale editor that is used to define sets of values and associate it to ACKTUS objects such as questions. In the example the ICF scale about the extent of impairment is shown.

Features that relates to the physical home environment (e.g., in terms of *spaces*) can be captured by the ACKTUS ontology at a generic level using the *building* class (Figure 4). Furthermore, smart objects can be considered as being *tools*, thus falling under this concept. Activities fall under the *Activities and Participation* class, following the ICF categorization as basic structure. Concerning ability to execute basic level tasks, such as handle doors, windows, etc., such features can be added to the activity class that concerns *mobility*. Limitations in ability that are caused by the characteristics of the physical environment need to be assessed in each individual case, since each individual's environment is unique. In [5] this was done by an occupational therapist. In our work presented in [6], two occupational therapists and a nurse assessed the older adult's life situation in our case scenario using the concepts (e.g., Figure 5) and scales (e.g., Figure 6) of assessment instruments typically used in their daily work.

The core ontology implements in addition to the user model the components of reasoning in the form of an argumentation framework. The argument interchange format (AIF) is used for visualising and typing different arguments and argumentation schemes [7]. Preferences can be expressed, e.g., about which knowledge sources to be used (clinical guidelines etc.).

2.1. Extending ACKTUS Applications with Adaptive Functionality in three Application Domains

A pilot project with the purpose to investigate needs and motives for tailored web interfaces to older adults for accomplishing activities that they wanted to perform generated a generic model of purposeful activity [8]. The main needs behind web-supported activities that were identified were i) feeling safe and secure (in particular as part of a social context), ii) having knowledge and control (e.g., keeping up with news), and iii) feeling good, healthy, engaged, active, having fun (e.g., performing activities in fields of interest). These themes were common to the participants and have been in our work so far considered generic enough to provide a sufficient base for personalisation in a home context. What is evident is that an important part of an input to the personalisation has to be the self-assessed view on activity performance and goal satisfaction. This part can also be achieved using ACKTUS [9]. However, the potential pro-active behaviour to invoke assessment and adapt to changing needs and abilities is lacking in the ACKTUS applications. This is also seen in application domains other than support systems for older adults. Therefore, we present ongoing research and some results in the following sub-sections.

2.1.1 Personalisation for Supporting Older Adults in Home Environments

Activity recognition is an important and interesting challenge to address in dealing with pro-active assessment and adaptation to changing user needs and abilities within a smart home. Activity recognition facilitates implicit data collection on: How everyday activities are performed quantitatively? For instance, how mandatory actions like turning off the stove after cooking and locking the door before leaving the home are performed? How often such mandatory actions are forgotten? What are the activities that are usually completed successfully? The list of activities that require further assistance and adaptation of the environment, etc. can be analysed. The everyday activities that are in focus for our work are higher level activities such as preparing breakfast that involves planning, organizing, decision making, allocating time, using physical/virtual equipments and taking part in social interaction. Events like forgetting to take the milk packet from the refrigerator and placing it on the dining table are modelled as actions (some could be mandatory actions). The activities are part of a 24-hours activity loop making it possible to monitor the activities and the organisation of activities performed by an individual.

We use a case scenario created based on the mentioned pilot project, which describes an older woman with some difficulties in her daily life and in her home environment [10]. We explore and describe the case study as four distinct but related activities: 1) initial assessment, 2) referral to physician, 3) determine interventions, 4) apply interventions in the daily life of the individual including a continuing assessment (and follow-up), and 5) a renewed assessment. A software architecture was proposed in [11], utilizing results from the Easy ADL project [12]. The conceptual design of an activity recognition system is being formed in a living laboratory home environment [13]. The living laboratory home environment to be used comprises of smart objects that are augmented with ambient intelligence technologies. The smart objects apart from their primary functionality are expected to provide additional virtual functionalities like sensing user interaction with the object, internal state changes caused by such interaction, exchanging collected sensor readings with trusted applications like the ACKTUS application, and providing access/control to virtual information through multiple modalities depending on the context. Further information about the underlying technological infrastructure: ecology of smart objects and a personal activity-centric middleware to manage those smart objects is available [14]. The data collected from sensors in smart objects will be interpreted into activities and matched to the semantics of the ACKTUS user model [11]. The idea is to synthesize the explicit knowledge and interaction designed by the domain experts with the implicit knowledge obtained from the smart home.

2.1.2 Personalisation for Supporting Mining Workers in their Work Environments

An ongoing project aims at supporting the workers in mining and mining-related work environments in valuing the risks of their work situation and create awareness in the individual about how he/she can decrease risks. The purpose is to evaluate the vibration, dust and skin related problems from the data collected when the worker uses support applications. The evaluation is done according to algorithms defined by experts in the field based on available knowledge. This could possibly lead to taking precautions such as wearing a mask if the air contains harmful components in the form of dust such as Lead (molecular formula: Pb). A prototype of a "vibration application" called *ArbetsVis* has been developed and will be evaluated in wider use during 2012. The application provides tailored calculations of risks and advices about what can be changed in the work context to reduce the vibration exposure [10].

In addition to vibration, mining workers are often exposed to dust of different types that may be hazardous depending on the level of exposure and the history of exposure. Another purpose is to provide the worker tailored assessment of risk and advice about preventing medical conditions due to the exposure. In a user model that can provide the base for such support, both information about the individual such as measures obtained by blood tests and/or pulmonary tests is important, as well as information about the presence of hazardous dust particles in the work environment. Time is also an important parameter in the calculations of the risk, e.g., time periods when the work tasks are conducted in dusty environments.

As part of this ongoing work, domain experts are modelling the explicit knowledge using ACKTUS. This includes both the medical experience they possess and the national guidelines governing what thresholds of exposure are allowed in work environments. In their work to identify key features of user models, they use the concept editor that allows them to categorise professions, machines, etc. so that the conceptual model of vital part of their work environment becomes familiar to an end user (Figure 4). Concepts that are already defined in ICF and reused in the project are concepts that relates to a physical environment such as *vibration*, *dust*, *chemical substance*, etc. The participating domain experts define the sub-categories of these generic concepts that are considered relevant and useful and form a more specific knowledge model.

In collaboration with one of the mining industries, a test repository is being created with measures of lung capacity and presence of lead in blood samples related to fictive individuals, similar to the database they use in the in-house medical service. Furthermore, it is investigated in which way measures of the presence of dust can be accomplished in the participating mining industries. Consequently, further analyses will be needed as part of future work to enable situated support to individuals based on such information.

2.1.3 Personalisation for Improving Diagnostic Reasoning in Health Professionals

There is an urgent need to provide medical professionals tools for verifying decisions, incorporating new research-based knowledge and monitoring a personal continuing medical education, as part of everyday practice in their meeting with patients. Studies have shown that 75% of dementia patients do not receive a dementia diagnosis at their first visit to a physician. Studies have also shown that introducing a clinical decision-support system for supporting dementia diagnosis can change work routines, increase teamwork and detect lack of knowledge in medical professionals [15-16].

Therefore, allowing for the medical professional to keep control both over his/her own (developing) "knowledge-base" as well as other actors' is an approach we apply in an ongoing research project. An additional motive is to provide methods for detecting emerging knowledge as interaction takes place, also at stages before the knowledge has become established as "evidence-based" medical knowledge. In this interaction, detecting patterns of reasoning in relation to individuals' knowledge and patient cases is central. For these purposes we intend to use repositories of data relating to patent cases and their physicians' investigations to explore the potentials in modelling support in the form of tailored, personalised dialogues [17].

2.2 Towards a Multi-Agent System for Personalisation

The agent paradigm is receiving increased focus and application to health care. There are particular benefits when using a multi-agent system (MAS) perspective on a distributed and collaborative use environment utilizing mixed-initiative functionality. Agents have autonomy to a certain extent to behave pro-active taking initiatives based on their repository of beliefs about its environment (e.g., [18]).

As described earlier, we take a persona and a case scenario as starting point for development of an ambient assisted living environment [11]. An analysis of the scenario generated an outline for a MAS design. We analysed the different dialogues exemplified in our scenario, interpreted them as *dialogue games*, and categorised them into three types of dialogues described by Walton [19]: *information seeking*, *inquiry* and *deliberation* dialogue. They differ by their purpose where information seeking aims at collecting information, inquiry dialogues aim at collaboratively create new knowledge and a deliberation dialogue aims at collaboratively decide upon an action to be performed.

For illustrating the dialogues in our use scenario in the development sessions with domain experts, we use an algorithm for executing the information seeking dialogues and for simulating the inquiry and deliberation dialogues. The algorithm only makes use of ACKTUS assessment protocols, their content, their associated rules and their consequents as dialogue flows structured by the domain experts in the modelling sessions. The dialogues were used in the sessions for the domain experts to evaluate the effects of their modelling [9].

A formal argumentation-based framework and the agent interaction protocols need to support the identified types of dialogues. Furthermore, we identified the following speech acts or moves to be performed by each agent: *open*, *ask*, *assert* and *close*. Consequently, the context-based inquiry dialogues proposed in [17] need to be extended to incorporate critical questions based on the ACKTUS critical questions and reasoning contexts to achieve the *ask* move. Initial results are presented in [20].

3. Related Work

Research on personalisation similar to the work presented in this paper has been carried out by Kadouche et al [5], with the *Semantic Matching Framework* (SMF) defining the user limitation capabilities and provides adapted process to personalise the service delivery. The core functionality of SMF is based on semantic matching between the user model and the environment model. The user model characterises user factors, for instance his name, his preferences and capacities, defined as user's attributes. The environment model describes environment factors it specifies devices (e.g., doors, windows, sensors, etc.), defined as "effectors", each effector contains a set of characteristics defined as environment's attributes, for instance: required force to open the door, the door size, etc. These factors are quantified to formalise the relation between the user's attributes and the environment's attributes which brings out the handicap situation for each user on his daily living activities [5]. The framework has been designed with experts in psychology and computer science [5].

A review of agents applied in health care is provided in [18] where the following examples are given. The *Context-aware Hospital Information System* (CHIS) is a MAS, which provides intelligence and proactive capabilities to healthcare environments furnished with ubiquitous computing and medical devices [21]. Some degree of personalisation is accomplished in this system. The interaction with the system is based on the user's permission in performing the activities. Another example of a MAS that supports personalisation is the project Health Care Services (HeCaSe2) [22], which offers healthcare services to the users (patients and practitioners). In this system, a single agent is associated to each user and the knowledge that is endowed to the user is based on the user's preferences and supports personalisation.

In the past decade, numerous studies on user modelling have been conducted by researchers [23]. The approaches have been discussed with respect to its methodology adopted for the user information. User profiles have been used in the healthcare domain to provide management and retrieval of person or patient's physiological data [24]. In a smart home, a cooking guide is a true effort towards the contextual rich dynamic proactive knowledge-based application. A proactive knowledge base is built from the sensors augmenting the objects in use, surrounding devices and user profiles. Sophisticated data mining algorithms, rule based mechanisms and user model learning techniques facilitate contextual awareness and adaptability towards the assistance and end user ambient support [25]. There are many techniques available to model an adaptive web system such as *feature-based modelling* and *stereotype-based modelling*. Feature-based modelling attempts to model specific features such as individual user's knowledge, interests, goals etc. The majority of modern web systems use the feature-based approach to represent and model information about the users such as Bayesian Student Models (BSM) in which the student features are modelled and parameters of the model are obtained [26]. An example of the stereotype-based model is the *User Modelling and Profiling Service* (UMPS), which provides the methodology for context dependent personalisation and adaptivity of applications and services in the Amigo environment [27]. Normally, a stereotype model ignores the features and uses the stereotype as a whole. In UPMS the user profiles are built first based on stereotypes and explicit user input, and in the second step these profiles are refined using the interaction/context history. User modelling, also known as user profiling is the key in the development of interactive software systems that are able to identify and to adjust themselves to the needs of particular user at every stage of use, irrespective of these user's knowledge whether they are experts or beginners.

User modelling has evolved from representation of groups of users using a certain system in certain conditions, to personalisation of these systems towards individual user's preferences and requirements. In the case of ambient intelligent systems personalisation applies to different system components and application domains, particularly to user interfaces, for example, graphical content, gesture and voice based and content presentation.

Personalisation depends upon the initial knowledge of the system about its potential users and the mechanism used to learn user's behaviour and preferences [28]. It is also dependent on the system's use context. Thus the system needs to be adaptable to the context as well. In the following sub-sections an overview is given of different real world applications making use of personalisation and user modelling, categorised by their purpose and context of use. We limit our focus to general assistance, smart home environments for supporting activities of daily living (ADL), intelligent tutoring systems and personalised systems for promoting health during work or leisure time.

3.1 Living Assistance Support Systems

Research related to living assistance broadens the scope of support systems. Such systems can be for indoor and outdoor assistance. Systems for indoor living assistance work in a well-defined locality, for example home, car, hospital or elderly care homes. Appropriate sensors, hardware and software can be installed for the suitable environment. The latter could be developed for activities such as shopping or travelling.

According to Nehmer et al [29], three types of services could be provided by a support system, namely, emergency treatment, autonomy enhancement and comfort. The emergency treatment plays the most important role as it aims at the early prediction of and recovery from critical conditions that might result in an emergency situation and safe detection and alert propagation of emergency situations. Examples are when patient suddenly falls, strokes etc. Autonomy enhancement services denote all services that make it possible to abandon previous manual care given by medical and social care personnel or relatives and replace it by appropriate support system. The best example for such assistance is a cooking assistance system for people with visual defects [29]. The appropriate sensors attached to a stove may enable them to cook safely, automatically detect if the stove is switched on for long time without any purpose and alert the user.

3.2 Assisted Living within a Smart Home

Personalisation and user modelling plays a central role in designing smart homes that enable elderly persons to live a longer and more independent life at home (in accordance with the concepts of successful ageing and assisted cognition). EU research programs like the Ambient Assisted Living (AAL) [30] are moving in this direction. Since life expectancy is increasing and more elderly people would populate the society in the future, efforts like AAL are necessary to provide a good quality life for the elderly. Elderly people are also prone to physical and cognitive disabilities that introduce a need to personalise and adapt their smart homes. Independent living means that the elderly person must be able to do basic activities of daily living like taking a shower, eating and dressing, and more complex activities of daily living like preparing a shopping list and cooking food. Activity recognition plays an important role in enabling independent living of elderly within their smart home. Everyday activities performed by different individuals within a home environment vary significantly, introducing a need to be aware of the user's identity, preferences, capabilities, likes and limitations. Also, there are variations in how certain activities are performed by an individual depending on the contextual conditions. Solutions that lack personalization affect the user experience provided by smart homes.

Smart homes for people with disabilities are an objective of many research efforts. The *Nursebot* project [31] is aimed at developing mobile robotic assistants that support elders in performing their activities of daily living (ADL) at home. The *SmartBo* project [32] focuses on elders with mobility impairments and cognitive disabilities while the *Gloucester Smart House* [33] focuses on people with dementia. An automated hand-washing assistant for people with dementia is being developed [34]. Automated health monitoring and anomaly detection for cognitively and physically challenged people using machine learning algorithms that can model their behaviour is also described in research [35]. AAL within the CareLab [36] has focused on monitoring and coaching elderly people to enable them to maintain an independent lifestyle by focusing on their feeling of safety, cognitive prosthesis and social interaction.

The easy ADL ecology [12] is a smart home that models a user's situation, activities and interaction possibilities with the long-term goal of providing support for mild-dementia patients. A human agent's body is used as a starting point to determine: a) the objects that are present in close proximity to the user useful in modelling their situation [37]; b) the objects that are manipulated by the user in modelling the activities performed [14]; and c) the interactive devices in the close proximity are useful for timely delivery of assisted living services [13].

A kitchen environment could be viewed as the heart of a home comprising of several household objects, home appliances and furniture facilitating numerous everyday activities with varying complexities. A smart refrigerator capable of presenting personalized information, intelligently evaluating its contents and informing about missing ingredients based on user profile or their potential future cooking activity is useful [38].

A bathroom is another important space in a home where self-care activities are performed. A shower cabin or a tub that adapts the water temperature based on user profile could be an automated service provided by the smart home. Brushing teeth, combing hair, shaving and applying make-up usually takes place inside a bathroom in front of the bathroom mirror. Assisted living services could be incorporated within the so-called "smart mirror" augmented with ambient intelligence technology that identifies the user (for instance, based on the tooth brush or the shaving razor used). The smart mirror apart from its original functionality of reflecting a person's image also presents personalised information relevant to that person. Examples include presenting the day schedule, news, melodies to remove stress and weather information. Additionally, by standing on a smart carpet in front of the mirror, the person's weight information and cardiovascular health status using existing medical data could also be presented on the smart mirror.

3.3 Intelligent Tutoring Systems

The majority of intelligent tutoring systems (ITS) focused on representing two types of domain knowledge: conceptual and procedural knowledge. A large class of ITS known as "tutors" focus on helping users solve educational problems, and thus rely on procedural knowledge of either problem solving or evaluation nature [39]. The use of conceptual knowledge is shared by almost all non- educational systems, which also focus on guiding the user to the most appropriate content.

The tutorial application iTutorial has been developed as part of an assisted living environment in the Share-it project [39]. Based on physical location and a set of individual-specific features (e.g., presence of memory impairment) the support is given in the form of detailed step-by-step instructions on how getting dressed, etc.

3.4 Intelligent Health Promotion Support Systems

Another application area attaining growing research interest is to enhance the work environment such that the risks, health hazards, physical and psychological strains could be analysed and predicted to facilitate a better work environment. An example of this is the application for stress diagnosis developed by Begum et al [8]. They present a computer-aided decision support system for analyzing and diagnosing stress-related disorders based upon finger temperature signals where the finger temperature measurement is taken using a temperature sensor to establish an individual's stress profile. The functionality of the system lies in solving a new problem case by using solution of solved cases from the past (case-based reasoning), which often require adaptation to find a suitable solution for the new case.

A wide range of applications target behaviour change in users to increase healthy living by persuading individuals to exercise, eat healthier, etc. One example is the work by Grasso and Erriques [40] where they used an ontology that included both person specific factors, activities and types of responses aimed at giving encouraging arguments to pursue a healthier lifestyle.

4. Summary and Future Work

ACKTUS as tool for modelling user information and for personalisation of support applications has been presented. Motivations to personalise support applications for a variety of daily living activities have been described: work and education (monitoring health in the mining work environment and for supporting dementia diagnosis), leisure and social activities (older adults' home environment), personal and home care (older adults' home environment).

For the purpose to provide an individual computer-based support in daily living for increasing autonomy, security, health, social inclusion and quality of life, a holistic view on the individual's situation needs to be adopted. The wishes, needs and abilities of the individual need to be assessed to optimize the design of the tailored support. Furthermore, the assessments need to be done continuously, to adjust the tailored support to changing needs, abilities, wishes and contextual factors. Therefore, we integrate *assessment* into a framework for developing and maintaining ambient assisted living with personalised support. This will be accomplished by using a combination of methods where the professional assessments done by health care professionals, play a key role.

An overview of the existing approaches presented in literature has been provided, which have been adopted for personalisation and user modelling for various domains. The purpose was to compare the applied methodologies with the ongoing effort to develop a generic framework for assessing ability and tailoring of support applications in the health domain (ACKTUS). Several related projects use similar technology and ambition, that is, to provide meaningful support in daily living to older adults, or tailored support for learning and decision-making. However, none of the approaches we found aimed at the holistic assessments that ACKTUS targets, neither allowing the domain professionals to conduct the modelling and application of an ontology and its content in user cases for evaluation purposes. Moreover, the novel approach to use a common core ontology for modelling support for various every day activities and knowledge domains, including work-related, was presented.

Ongoing and future work includes among other tasks the investigation of how data obtained in a home or work environment can be refined into information that can be interpreted as qualitative knowledge about an individual's ability to perform activity, and providing situated risk assessments. In this context, an important research question is how to create a realistic mapping between existing scales for measuring user ability (e.g., ICF-scales, ADL-scales, the AAIMA-protocol [8]) and the sensor information obtained from smart objects. The determination of what state changes to objects are important, what does such state changes mean when interpreted according to the existing scales, how to model everyday activities such that the activity recognition models are comparable to the models described by the existing scales, are important aspects to dig deeper in adapting the existing living laboratory home environment to a personalised environment for conducting pro-active assessment of older adults' abilities.

References

1. Kobsa, A. Generic user modeling systems. User modeling and user adapted interaction, 11, 49–63, (2001).
2. Alatalo, T., and Siponen, M. T. Addressing the personalization paradox in the development of electronic commerce systems. In Post-proceedings of the EBusiness Research Forum (eBRF), Tampere, Finland, (2001).
3. Lindgren H, Winnberg P. A Model for Interaction Design of Personalised Knowledge Systems in the Health Domain. In M. Szomszor and P. Kostkova (Eds.): E-Health 2010, LNICST 69, pp. 235–242, Springer Verlag (2011).
4. Nardi, B. (ed). Context and Consciousness: Activity Theory and Human-Computer Interaction. MIT Press, (1996).
5. Kadouche, R., Abdulrazak, B, Giroux, S., Mokhtari, M. Personalization and Multi-user management in Smart Homes for disabled people. International Journal of Smart Home (IJSH), Vol.3, No.1, 39-48 (2009).
6. Lindgren, H., Nilsson, I. Towards User-Authored Agent Dialogues for Assessment in Personalised Ambient Assisted Living. (Submitted)
7. Wilmott S., Vreeswijk G., Chesñevar C., South M., McGinnis J., Modgil S., Rahwahn I., Reed C., Simari G. Towards an Argument Interchange Format for Multiagent Systems. In Proc. of the 3rd. Intl. Workshop on Argumentation and Multiagent Systems (ArgMAS 2006), Japan, May (2006).
8. Lindgren, H. Personalisation of Internet-Mediated Activity Support Systems in the Rehabilitation of Older Adults – A Pilot Study. In: proc Personalisation for e-Health 2009, pp. 20-27, Verona, (2009).
9. Lindgren, H., Nilsson, I. Towards User-Authored Agent Dialogues for Assessment in Personalised Ambient Assisted Living. (Submitted)
10. Lindgren, H. Persuasive Technology Composed by Domain Experts for Improving Health in the Mining Industry. (submitted)
11. Lindgren H, Surie D, Nilsson I. Agent-Supported Assessment for Adaptive and Personalized Ambient Assisted Living. In: Juan M. Corchado, Javier Bajo, Kasper Hallenborg, Paulina Golinska and Rafael Corchuelo (Eds.)

Trends in Practical Applications of Agents and Multiagent Systems, pp. 23-32. Springer (2011).
12. Surie, D., Janlert, L-E., Pederson, T., Roy, D. Egocentric interaction as a tool for designing ambient ecologies – The case of the easy ADL ecology. Journal of Pervasive and Mobile Computing, Special Issue on Ambient Ecologies, Elsevier, January (2012).
13. Surie, D. Egocentric interaction for Ambient Intelligence. PhD Thesis, Department of Computing Science, Umeå University, Sweden, (2012).
14. Surie, D., Pederson, T., Lagriffoul, F., Janlert, L.-E., and Sjölie, D. Activity Recognition using an Egocentric Perspective of Everyday Objects. In Proceedings of the IFIP International Conference on Ubiquitous Intelligence and Computing, Hong Kong, China, Springer LNCS 4611, pp. 246-257, (2007).
15. Lindgren H. Towards personalized decision support in the dementia domain based on clinical practice guidelines. User Modeling and User-Adapted Interaction 21(4):377-406 (2011).
16. Lindgren, H. Limitations in Physicians' knowledge when assessing dementia diseases – an evaluation study of a decision-support system. Stud Health Technol Inform 169;120-124, IOS Press (2011).
17. Lindgren H. Towards Context-Based Inquiry Dialogues for Personalized Interaction. In: Yves Demazeau, Michal Pechoucek, Juan M. Corchado and Javier Bajo (Eds.) PAAMS 2011. Advances in Intelligent and Soft Computing 88, pp. 151-161, Springer (2011).
18. Isern, D., Sánchez, D., and Moreno, A. Agents applied in health care: A review. I. J. Medical Informatics 79(3): 145-166 (2010).
19. Walton, D. Argumentation Schemes for Presumptive Reasoning. Mahwah, N.J.: Erlbaum (1996).
20. Lindgren H., Yekeh F., Baskar J., Yan C. Agent-Supported Assessment for Personalized Ambient Assisted Living. To appear in: Proceedings of Agents for Health Care workshop (A2HC 2012) (2012).
21. Rodríguez, M.D., Favela, J., Preciado, A., Vizcaíno, A. Agent-based ambient intelligence for healthcare. AI Communications 18 (3) 201–216 (2005).
22. Isern, D., Sánchez, D., Moreno, A. HeCaSe2: a multi-agent ontology-driven guideline enactment engine, in: H.-D. Burkhard, G. Lindemann, R. Verbrugge, L.Z. Varga (Eds.), Proceedings of the 5th International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2007, vol. 4696 of LNAI, Springer Verlag, Leipzig, Germany, pp. 322–324 (2007).
23. Harvey, C. F., Smith, P., and Lund, P. Providing a networked future for interpersonal information retrieval: InfoVine and user modeling. Interacting with Computers, 10, 195- 212 (1998).
24. Eslami, M.Z., Zarghami, A., Sapkota, B., and Sinderen, M.V., Personalized Service Creation by Non-technical Users in the Homecare Domain, In Proceedings of the International Conference on Ambient Systems, Networks and Technologies (ANT 2011), Ontario, Canada (2011).
25. Vildjiounaite, E., Kantorovitch, J., Niskanen, I., Towards the user confidence in sensor-rich interactive application scenarios. Accepted to workshop on Semantic Models for Adaptive Interactive Systems (SEMAIS 2011) at IUI 2011, Palo Alto, USA (2011).
26. Friedewald, M., Costa, O. C., Science and Technology Roadmapping: Ambient Intelligence in Everyday Life (AmI@Life). June (2003).
27. Magerkurth, C., Etter, R., Janse, M., Kela, J. Kocsis, O., Ramparany, F. An intelligent user service architecture for networked home environments, In: 2nd International Conference on Intelligent Environments (2006).
28. Khedr, M., Karmouch, A. ACAN-Ah hoc Context Aware Network. IEEE CCECE'02, Winnipeg, Canada (2002).
29. Nehmer, J., Becker, M., Karshmer, A., and Lamm, R. Living assistance systems: an ambient intelligence approach. In Proceeding of the 28th international Conference on Software Engineering. ICSE '06. ACM Press, New York, NY, 43-50, Shanghai, China, May 20 - 28 (2006).
30. AAL.: The Ambient Assisted Living (AAL) Joint Programme, Available online at http://www.aal-europe.eu/ (accessed 20th February 2012).
31. Pollack, M.E., Brown, L., Colbry, D., Orosz, C., Peintner, B., Ramakrishnan, S., Engberg, S., Matthews, J., D.-J., J., McCarthy, C., Thrun, S., Montemerlo, M., Pineau, J., Roy, N. Pearl: A Mobile Robotic Assistant for the Elderly. AAAI Workshop on Automation as Caregiver, (2002).
32. Elger, G., Furugen, B.: "SmartBo" – An ICT and computer-based demonstration home for disabled people, Improving the Quality of Life for the European Citizen. IOS Press, (1998).
33. Adlam, T., Orpwood, R. Taking the Gloucester Smart House from the Laboratory to the Living Room. In proc. The 3rd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications, (2004).
34. Hoey, J., Poupart, P., Bertoldi, A.V., Craig, T., Boutilier, C. and Mihailidis, A. Automated handwashing assistance for persons with dementia using video and a partially observable Markov decision process. Computer Vision and Image Understanding, Special Issue on Intelligent Vision Systems, Volume 114, pp. 503-51, (2010).
35. Jakkula, V., Cook, D. Anomaly detection using temporal data mining in a smart home environment. Methods of Information in Medicine, Volume 47, pp. 70-5, (2008).
36. Ruyter, B.D., Pelgrim, E. Ambient assisted-living research in carelab, Interactions, Volume 14, pp. 30-33, (2007).
37. Pederson, T., Janlert, L-E., Surie, D. A Situative Space Model for Mobile Mixed-Reality Computing. IEEE Pervasive Computing, Vol. 10, Issue 4, (2011).
38. Brusilovsky P., Kobsa A., and Nejdl W., (Eds.): The Adaptive Web, LNCS 4321, pp. 3 – 53, (2007).
39. Rubio, C., Annicchiarico, R., Barrue, C., Cortes, U., Sanchez-Marre, M., Caltagirone, C. iTutorials for the Aid of Cognitively Impaired Elderly Population. eHealth 2010: 303-310 (2010).
40. Erriquez E. and Grasso, F. Generation of Personalised Advisory Messages: an Ontology Based Approach. 21th IEEE International Symposium on Computer-Based Medical Systems, Jyvaskyla, Finland, 19-21, p. 437-442, IEEE press (2008).
41. Begum, S., Ahmed, M.U., Funk, P., Xiong, N., Schéele, B.V.. A Case-Based Decision Support System for Individual Stress Diagnosis Using Fuzzy Similarity Matching. Computational Intelligence (CI), vol 25, nr 3, p180-195(16), Blackwell Publishing, August, (2009).

The 27th annual workshop of the Swedish Artificial Intelligence Society (SAIS),
14–15 May 2012, Örebro, Sweden
Published by Linköping University Electronic Press
http://www.ep.liu.se/ecp_home/index.en.aspx?issue=071

Managing Inconsistent Possibilistic Knowledge Bases by An Argumentation Approach

Juan Carlos Nieves[1], Mauricio Osorio[2], Helena Lindgren[1]
[1]Department of Computer Science
Umeå University
SE-901 87, Umeå, Sweden
Email: {jcnieves,helena}@cs.umu.de

[2]Universidad de las Américas - Puebla
Dept. de Actuaría y Matemáticas
Sta. Catarina Mártir, Cholula, Puebla, 72820 México
Email: osoriomauri@gmail.com

Abstract

Inconsistent knowledge bases usually are regarded as an epistemic hell that have to be avoided at all costs. However, many times it is difficult or impossible to stay away of managing inconsistent knowledge bases. In this paper, we introduce an argumentation-based approach in order to manage inconsistent possibilistic knowledge bases. This approach will be flexible enough for managing inconsistent possibilistic models and the non-existence of possibilistic models of a possibilistic logic program.

1 Introduction

One of the purposes of argumentation theory is to provide tools for supporting decisions. For instance, argumentation theory is able to suggest arguments in favour a decision. Usually argumentation theory is adequate for supporting decisions in scenarios where the information is inconsistent and incomplete [13]. Indeed, an interesting feature of argumentation theory inference is that it is able to manage inconsistent information in a natural way.

In [8, 10, 12], a possibilistic framework for reasoning under uncertainty was proposed. This framework is a combination between Answer Set Programming (ASP) and Possibilistic Logic [4]. Possibilistic Logic is based on possibilistic theory where at the mathematical level, degrees of possibility and necessity are closely related to fuzzy sets. Thanks to the natural properties of possibilistic logic and ASP, this approach allows to deal with reasoning that is at the same time non-monotonic and uncertain. The expressiveness of this approach is rich enough for capturing sophisticated domains such as the medical domain [10] and river basin systems [1]. However, a common problem in these domains is that they are inconsistent and incomplete. It is not difficult to find scenarios where the specification of them is inconsistent. We take as example in this paper the observation made in [7] that some medical diagnoses in the dementia domain are based on incomplete and inconsistent information. For instance, in order to assess the diagnosis Alzheimer's disease the presence of episodic memory dysfunction is required according to diagnostic criteria. In the study presented in [7] this assessment was sometimes based on missing information and sometimes based on the absence of memory deficit.

In this paper, we explore an argumentation approach which consider any possibilistic logic programming semantics [8, 10, 12] for inferring information from any consistent or inconsistent possibilistic logic program.

Usually the inference in argumentation theory can be regarded by 4 steps:

1. *Argument construction*, based on a knowledge base;
2. *Argument valuation*, the assignment of a weight to arguments;
3. *Argumentation interaction*, the identification of conflicts between the arguments; and
4. *Argumentation status evaluation*, for deciding the winning or justified arguments.

In this paper essentially we will define a strategy for building possibilistic arguments by considering a possibilistic logic program. A possibilistic argument will have a standard structure (as in argumentation theory) and will be inferred by possibilistic logic programming semantics, such as the possibilistic stable semantics [8], the possibilistic answer set semantics [10] or the possibilistic pstable semantics [12], of the given possibilistic knowledge base.

In order to manage the conflict that could appear between possibilistic arguments, we will instantiate the argumentation framework structure introduce by Dung [5] in terms of possibilistic arguments. This instantiation will allow us to use argumentation semantics *e.g.,* preferred semantics, in order to infer justified arguments.

The rest of this paper is divided as follows: in Section 2, we will present some basic definitions of the logic programming syntaxis considered. In Section 3, we will present the definition of a possibilistic argument. After that, in Section 4, we formalize the interaction of possibilistic arguments. In Section 5, we define our approach for evaluating possibilistic arguments (due to lack of space all *proofs* will be omitted). Finally, in the last section, we outline our conclusions and future work.

2 Background

In this section, the syntaxis of the logic programs considered in the paper is briefly presented.

2.1 Non-Possibilistic Logic Programs

An *atom* is a propositional symbol such $a_0, a_1, \ldots$ and *an extended atom* is a negated atom such as $\neg a_0, \neg a_1, \ldots$. We will use the concept of atom without paying attention if it is an extended atom or not. Notice that there are complementary atoms such that the complement of an atom a is defined as $\widetilde{a} = \neg a$ and $\widetilde{\neg a} = a$. The negation sign $\neg$ is regarded as the so called *strong negation* by the ASP's literature and the negation *not* as the *negation as failure* [2]. A *literal* is an atom, a, or the negation of an atom *not* a. Given a set of atoms $\{a_1, ..., a_n\}$, we write *not* $\{a_1, ..., a_n\}$ to denote the set of literals $\{not\ a_1, ..., not\ a_n\}$. An *extended disjunctive clause*, C, is denoted:

$$a_1 \vee \ldots \vee a_m \leftarrow a_1, \ldots, a_j, not\ a_{j+1}, \ldots, not\ a_n$$

where $m \geq 0$, $n \geq 0$, each a_i is an atom. When $n = 0$ and $m > 0$, the clause is an abbreviation of $a_1 \vee \ldots \vee a_m$. When $m = 0$ the clause is an abbreviation of $\bot \leftarrow a_1, \ldots, a_n$ such that $\bot$ is the proposition symbol that always evaluates to false. Clauses of this form are called constraints (the rest, non-constraint clauses). An extended disjunctive program P is a finite set of extended disjunctive clauses. By $\mathcal{L}_P$, we denote the set of atoms in the language of P.

We denote an extended disjunctive clause C by $\mathcal{A} \leftarrow \mathcal{B}^+$, *not* $\mathcal{B}^-$, where $\mathcal{A}$ contains all the head atoms, $\mathcal{B}^+$ contains all the positive body atoms and $\mathcal{B}^-$ contains all the negative body atoms. When $\mathcal{B}^- = \emptyset$, the clause is called positive disjunctive clause. A set of positive disjunctive clauses is called a positive disjunctive logic program. When $\mathcal{A}$ is a singleton set, the clause can be regarded as a normal clause. A normal logic program is a finite set of normal clauses. Finally, when $\mathcal{A}$ is a singleton set and $\mathcal{B}^- = \emptyset$, the clause can be also regarded as a definite clause. A finite set of definite clauses is called a definite logic program.

We will manage the strong negation ($\neg$), in our logic programs, as it is done in ASP [2]. Basically, each negative atom $\neg a$ is replaced by a new atom symbol a' which does not appear in the language of the program.

For instance, let P be the normal program:

$$\begin{array}{ll} a & \leftarrow q. \\ \neg q & \leftarrow r. \\ q. & \\ r. & \end{array}$$

Then replacing each negative atom by a new atom symbol, we will have:

$$\begin{array}{ll} a & \leftarrow q. \\ q' & \leftarrow r. \\ q. & \\ r. & \end{array}$$

Usually it is added a constraint of the form $\bot \leftarrow a, a'$, we will omit this constraint in order to allow complementary atoms in the models of a program. However the user could add this constraint without losing generality.

2.2 Possibilistic Logic Programs

A *possibilistic atom* is a pair $p = (a, q) \in \mathcal{A} \times Q$, where $\mathcal{A}$ is a finite set of atoms and $(Q, \leq)$ is a lattice (in all the paper, we will consider only finite lattices). We apply the projection $*$ as follows: $p^* = a$. Given a set of possibilistic atoms S, we define the generalization of $*$ over S as follows: $S^* = \{p^* | p \in S\}$. Given a lattice $(Q, \leq)$ and $S \subseteq Q$, $LUB(S)$ denotes the least upper bound of S and $GLB(S)$ denotes the greatest lower bound of S. Given a finite set of atoms $\mathcal{A}$ and a lattice (Q,$\leq$), we denote by $\mathcal{PS}$ the power set $2^{\mathcal{A} \times Q}$. A possibilistic disjunctive clause is of the form:

$$r = (\alpha : \mathcal{A} \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^-)$$

where $\alpha \in Q$. The projection $*$ for a possibilistic clause is $r^* = \mathcal{A} \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^-$. $n(r) = \alpha$ is a necessity degree representing the certainty level of the information described by r. A possibilistic constraint is of the form:

$$c = (TOP_Q :\ \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^-)$$

where TOP_Q is the top of the lattice $(Q, \leq)$. As in possibilistic clauses, the projection $*$ for a possibilistic constraint is : $c^* = \ \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^-$. A possibilistic disjunctive logic program P is a tuple of the form $\langle (Q, \leq), N \rangle$, where N is a finite set of possibilistic disjunctive clauses and possibilistic constraints. The generalization of $*$ over P is as follows: $P^* = \{r^* | r \in N\}$. Notice that P^* is an extended disjunctive program. When P^* is a normal program, P is called a possibilistic normal program. Also when P^* is a positive disjunctive program, P is called a possibilistic positive logic program. A given set of possibilistic disjunctive clauses $\{\gamma, \ldots, \gamma\}$ is also represented as $\{\gamma; \ldots; \gamma\}$ to avoid ambiguities with the use of comma in the body of the clauses.

The semantics of possibilistic logic programs is captured by *possibilistic logic programming semantics*. A possibilistic logic programming semantics is a mapping from the class of all the possibilistic programs into $2^{\mathcal{PS}}$. In the literature, we can find several approaches for capturing the semantics of possibilistic logic programs [8, 10, 12].

3 Building possibilistic arguments

As we commented in Section 1, the first step in the inference process in argumentation theory is the construction of arguments. Hence, in this section, we start by defining how to build possibilistic arguments from a possibilistic program.

A possibilistic argument can be constructed by considering any possibilistic logic programming semantics *i.e.* the possibilistic stable semantics, the possibilistic answer set semantics, the possibilistic pstable semantics. Since one can consider the skeptical and credulous versions of possibilistic semantics as the possibilistic answer set semantics and the possibilistic pstable semantics, we will define two kinds of possibilistic arguments: *brave possibilistic arguments* and *cautious possibilistic arguments*[1].

Definition 1 (Possibilistic Arguments) *Let $P = \langle (\mathcal{Q}, \leq), N \rangle$ be a possibilistic logic program. A possibilistic argument Arg w.r.t. P is a tuple of the form $Arg = \langle Claim, Support, \alpha \rangle$ such that the following conditions hold:*

1. *$Support \subseteq N$.*
2. *Support is minimal w.r.t. set inclusion.*
3. *$\exists\ M \in S(Support)$ such that $(Claim, \alpha) \in M$ (in this case the possibilistic arguments Arg is called* brave*. When the existent quantified $\exists$ is changed by the for all quantified $\forall$, the possibilistic arguments Arg is called* cautious*).*

S is any possibilistic logic programming semantics. Brave-$\mathcal{ARG}_P^S$ gathers all the brave possibilistic arguments which can be constructed from P and the possibilistic logic programming semantics

[1] We adjectives of *brave* and *cautious* are motivated by the definitions of brave reasoning and cautious reasoning [6]

S. Cautious-$\mathcal{ARG}_P^S$ gathers all the cautious possibilistic arguments which can be constructed from P and the possibilistic logic programming semantics S.

In order to simplify the following definition, $\mathcal{ARG}_P^S$ will denote any set of possibilistic arguments constructed from P and based on the possibilistic logic programming semantics S. Observe that Definition 1 considers the first two steps of the inference in argumentation: *argumentation construction* and *argumentation valuation.*

Remark 1 *Before to follow on, we want to point out to the reader that to build an argument Arg with conclusion a from a program P, it does not mean that a is a correct conclusion of the whole program P. The acceptance of the conclusions will depend on the interaction of all the possibilistic arguments that one can build from P and the pattern of selection of arguments (argumentation semantics) that one uses for fixing the status of the arguments.*

4 Interaction between possibilistic arguments

Once we have defined how to build possibilistic arguments, we require to define how the possibilistic arguments will interact. In other words, we will define the cases when two possibilistic arguments will be in a conflict and then to define which arguments will be considered accepted according to a pattern of selection (argumentation semantics).

Definition 2 *Let Arg_1 and Arg_2 be two possibilistic arguments such that $Arg_1 = \langle Claim_1, Support_1, \alpha_1\rangle$ and $Arg_2 = \langle Claim_2, Support_2, \alpha_2\rangle$. We say that Arg_1 attacks Arg_2 if one of the following conditions hold:*

i) $Claim_1 = l$, $Claim_2 = \widetilde{l}$ *and* $\alpha_1 \geq \alpha_2$.

ii) $\exists(q : l \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^-) \in Support_2$ *such that* $\widetilde{Claim_1} \in \mathcal{B}^+$ *and* $\alpha_1 \geq \alpha_2$.

iii) $\exists(q : l \leftarrow \mathcal{B}^+,\ not\ \mathcal{B}^-) \in Support_2$ *and* $Claim_1 \in \mathcal{B}^-$.

Let us observe that only the first two conditions of the attack's definition have restrictions with respect to α's values. In the last condition, the attack relation is motivated by the fact that the claim of Arg_1 was assumed as false (by using *negation as failure*) in the support of Arg_2. Hence, the third condition has assigning less priority to literals which are negated by negation as failure which supporting a claim.

In Example 1 we will consider diagnosis of Alzheimer's disease (a), which is the most common dementia disease. Episodic memory dysfunction is a key finding (b), and we assume that we have two inconsistent pieces of information regarding memory. Furthermore, we know also that a large part of patients with Alzheimer's disease develop behavior and psychiatric symptoms (BPSD) (c).

Example 1 *Let P be the following possibilistic logic program:*

$0.1 : b \leftarrow \top.$	$0.8 : \neg b \leftarrow \top.$
$0.9 : \neg a \leftarrow\ not\ b.$	$0.5 : c \leftarrow a.$
	$0.5 : a \leftarrow \top.$

As we can see, one can construct at least the following four possibilistic arguments by considering any reasonable possibilistic argumentation semantics:

$$Arg_1 = \langle b, \{0.1 : b \leftarrow \top\}, 0.1\rangle$$
$$Arg_2 = \langle \neg b, \{0.8 : \neg b \leftarrow \top\}, 0.8\rangle$$
$$Arg_3 = \langle \neg a, \{0.9 : \neg a \leftarrow\ not\ b\}, 0.9\rangle$$
$$Arg_4 = \langle c, \{0.5 : c \leftarrow a; 0.5 : a \leftarrow \top\}, 0.5\rangle$$

By instantiating Definition 2, one can identify the following conflicts between these possibilistic arguments:

Arg_2 attacks Arg_1 by condition i).
Arg_3 attacks Arg_4 by condition ii).
Arg_1 attacks Arg_3 by condition iii).

Observe that Arg_1 does not attack Arg_2 because Arg_1's possibilistic degree is less than Arg_2's possibilistic degree.

Once the relationship between possibilistic arguments has been identified, we need to evaluate these relationships. In our example the finding that memory dysfunction is absent attacks the finding that it is possibly present, and the claim that there is BPSD symptoms is attacked by the claim that there is no Alzheimer's disease to cause BPSD. Furthermore, the argument that there is memory dysfunction attacks the claim that there is no Alzheimer's disease.

This evaluation process will be the objective of the next section.

5 Argumentation status evaluation

The evaluation of the interaction between arguments is an important step in the inference of argumentation. In argumentation literature, there are several approaches [3, 13] in order to select coherent points of view from a set of arguments in conflict. In our case, we will follow Dung's argumentation style [5]. This approach is based on the structure called *argumentation framework*. We will generalize the concept of argumentation framework into the concept of *possibilistic argumentation framework*.

Definition 3 *Given a possibilistic logic program, a possibilistic argumentation framework AF w.r.t. P is the tuple* $AF_P^S = \langle \mathcal{ARG}_P^S, Attacks \rangle$, *where Attacks contains the relations of attack between the arguments of* $\mathcal{ARG}_P$.

We are essentially instantiating Dung's argumentation approach into possibilistic arguments.

Example 2 *Let us go back to Example 1. As we saw, one can construct four possibilistic arguments from P; this means that* $\mathcal{ARG}_P = \{Arg_1, Arg_2, Arg_3, Arg_4\}$ *and the relations of attacks between these arguments are:* Arg_2 *attacks* Arg_1, Arg_3 *attacks* Arg_4, Arg_1 *attacks* Arg_3. *Hence, we have the following possibilistic argumentation framework:*

$$AF_P = \langle \{Arg_1, Arg_2, Arg_3, Arg_4\}, \{(Arg_2, Arg_1), (Arg_3, Arg_4), (Arg_1, Arg_3)\} \rangle$$

Once we have instantiated a possibilistic program P into a possibilistic argumentation framework AF_P, we can apply an argumentation semantics to AF_P in order to infer information from P.

In argumentation literature, we find that the most accepted argumentation semantics are the *grounded, stable and preferred semantics* suggested by Dung in [5]. The objective of these semantics is to select subsets of arguments from a set of arguments such that these subsets of arguments represent coherent points of view from a conflict. By coherent point of view, we mean that a set of arguments inferred by an argumentation semantics must be *consistent* and moreover it must be *a defendable position* in a conflict of opinions.

In order to study a relationship between the argumentation inference and the inference of some possibilistic logic programming semantics, let us define the projection ϕ which is a relation from $\mathcal{ARG}_P$ into $2^{\mathcal{PS}}$ such that given a set of possibilistic arguments $\mathcal{ARG}$, $\phi(\mathcal{ARG}) = \{(a, \alpha) | \langle a, Support, \alpha \rangle \in \mathcal{ARG}\}$.

Example 3 *Let us consider the possibilistic logic program of Example 1 and the argumentation framework of Example 2. As we saw the possibilistic argumentation framework:*

$$AF_P = \langle \{Arg_1, Arg_2, Arg_3, Arg_4\}, \{(Arg_2, Arg_1), (Arg_3, Arg_4), (Arg_1, Arg_3)\} \rangle$$

is an instantiation of the possibilistic logic program P:

$0.1 : b \leftarrow \top.$ $\quad 0.8 : \neg b \leftarrow \top.$
$0.9 : \neg a \leftarrow \; not \; b.$ $\quad 0.5 : c \leftarrow a.$
$0.5 : a \leftarrow \top.$

In order to illustrate the relation between the inference of argumentation theory and possibilistic logic programming semantics, let us consider the possibilistic answer set semantics defined in [9].

It is easy to see that the only possibilistic answer set S of the program P is the inconsistent set $\{(b, 0.1), (\neg b, 0.8), (c, 0.5)\}$. *On the other hand, by applying an argumentation semantics as the preferred semantics to* AF_P, *we can see that the only preferred extension PE of* AF_P *is* $\{Arg_2, Arg_3\}$. *This means that the argumentation inference based on the preferred semantics infers from P the possibilistic set* $\phi(PE) = \{(\neg b, 0.8), (\neg a, 0.9)\}$. *This means, interpreted by our medical example, that in this case there is no episodic memory dysfunction nor Alzheimer's disease present, which would be an accurate conclusion according to medical diagnostic criteria. Observe that the argumentation inference is removing the inconsistency of the set of possibilistic atoms which can be inferred from P. It is worth remember that the possibilistic argumentation framework* AF_P *can be constructed from P* *by considering the possibilistic answer set semantics.*

Now let us formalize some important properties *w.r.t.* our argumentation inference. The first property that we can identify is that whenever we apply our possibilistic-based argumentation inference based on an argumentation semantics which satisfies the basic property of conflict-freeness, we will infer consistent information. It is worth mentioning that an argumentation semantics S_{arg} satisfies the property of conflict-freeness, if given an argumentation framework AF then for all $E \in S_{arg}(AF)$, E does not contain two arguments a and b such that a attacks b.

Proposition 1 *Let S be a possibilistic logic programming semantics, S_{arg} be an argumentation semantics which satisfies the property of conflict-freeness and P be a possibilistic logic program. If $E \in S_{arg}(AF_P^S)$ then $\phi(E)$ is a consistent set of possibilistic atoms.*

Observe that this proposition is relevant whenever the information inferred from P by using the possibilistic logic programming semantics S is inconsistent. However, if the information inferred from a possibilistic logic program by a possibilistic logic programming semantics is consistent, there are cases where by considering our argumentation inference based on argumentation semantics as the preferred semantics, we are able to infer more information than applying the inference based possibilistic logic semantics (due to lack of space we do not illustrate this situation).

Proposition 2 *Let S be a possibilistic logic programming semantics, S_{arg} be the preferred semantics and P be a possibilistic logic program, If $E \in S(P)$, then there exists $E' \in S_{arg}(AF_P^S)$ such that $E = \phi(E')$.*

Another kind of inconsistency that can occur in possibilistic logic programs is the non-existent of models inferred from a possibilistic logic program by a possibilistic logic programming semantics *i.e.* the possibilistic answer set semantics and the possibilistic pstable semantics. For instance, let us consider the following program P_{inc}:

$0.3 : a \leftarrow \; not\ b.$
$0.5 : b \leftarrow \; not\ c.$
$0.6 : c \leftarrow \; not\ a.$

This program has neither possibilistic answer sets nor possibilistic pstable models. However, this kind of inconsistency can be managed by our argumentation-based inference. For instance, by considering either the possibilistic answer set semantics or the possibilistic pstable semantics, we have the following set of possibilistic arguments:

$$\mathcal{ARG}_{P_{inc}} =$$
$$\{Arg_1 = \langle a, \{0.3 : a \leftarrow \; not\ b\}, 0.3\rangle,$$
$$Arg_2 = \langle b, \{0.5 : b \leftarrow \; not\ c\}, 0.5\rangle,$$
$$Arg_3 = \langle c, \{0.6 : c \leftarrow \; not\ a\}, 0.6\rangle \; \}$$

Hence, we can define the following possibilistic argumentation framework: $AF_{P_{inc}} = \langle\{Arg_1, Arg_2, Arg_3\}, \; \{(Arg_1, Arg_3), \; (Arg_3, Arg_2), (Arg_2, Arg_1)\}$.

Observe that if we apply an argumentation semantics based on admissible sets *e.g.,* the preferred semantics to $AF_{P_{inc}}$, we will not be able to infer any set of possibilistic atoms from P_{inc}. For these cases, we require to consider argumentation semantics which are not based on admissible sets. For instance, if we consider the argumentation semantics $MM^{*^r}_{Arg}$ presented in [11], we can see that $MM^{*^r}_{Arg}(AF_{P_{inc}})$ has three extensions: $\{\{Arg_1\}, \{Arg_2\}, \{Arg_3\}\}$. This means that $MM^{*^r}_{Arg}$ is suggesting that one can infer the following three sets of possibilistic atoms from P_{inc}:$\phi(Arg_1) = \{(a, 0.3)\}$, $\phi(Arg_2) = \{(b, 0.5)\}$ and $\phi(Arg_3) = \{(c, 0.6)\}$.

6 Concluding remarks

In this paper, we defined a possibilistic-based argumentation approach based on: i. the inference of possibilistic logic programming semantics and ii. Dung's argumentation semantics style.

This approach inherits all the expressiveness of the possibilistic logic programs [8, 10, 12] and offers some natural mechanisms for dealing with reasoning under inconsistent information. In fact, this approach does not require to apply *cuts* to an inconsistent possibilistic knowledge base, as it is done in possibilistic logic programming, in order to manage the non-existence of possibilistic models. Another interesting property of our approach is that any set of possibilistic atoms inferred by the possibilistic-based argumentation inference will be consistent.

Acknowledgement

This research is partly funded by VINNOVA (Sweden's innovation agency) and the Swedish Brain Power.

References

[1] Montse Aulinas, Juan Carlos Nieves, Manel Poch, and Ulises Cortés. Supporting Decision Making in Urban Wastewater Systems using a Knowledge-Based Approach. *Environmental Modelling and Software*, 26(5):562–572, 2011.

[2] Chitta Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press, Cambridge, 2003.

[3] Carlos Iván Chesñevar, Ana Gabriela Maguitman, and Ronald Prescott Loui. Logical models of argument. *ACM Comput. Surv.*, 32(4):337–383, 2000.

[4] Didier Dubois, Jérôme Lang, and Henri Prade. Possibilistic logic. In Dov Gabbay, Christopher J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning*, pages 439–513. Oxford University Press, Oxford, 1994.

[5] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

[6] Nicola Leone, Gerard Pfeifer, Wolfgang, Thomas Either, Georg Gottlob abd Simona Perri, and Francesco Scarcello. The DLV System for Knowledge Representation and Reasoning. *ACM Transaction on Computational Logic*, 7(3):499–562, July 2006.

[7] Helena Lindgren. Limitations in physicians knowledge when assessing dementia diseases - an evaluation study of a decision-support system. *Studies In Health Technology And Informatics*, 169:120–124, 2011.

[8] Pascal Nicolas, Laurent Garcia, Igor Stéphan, and Claire Lefèvre. Possibilistic Uncertainty Handling for Answer Set Programming. *Annals of Mathematics and Artificial Intelligence*, 47(1-2):139–181, June 2006.

[9] Juan Carlos Nieves, Mauricio Osorio, and Ulises Cortés. Semantics for possibilistic disjunctive programs. In Stefania Costantini and Richard Watson, editors, *Answer Set Programming: Advances in Theory and Implementation*, pages 271–284, 2007.

[10] Juan Carlos Nieves, Mauricio Osorio, and Ulises Cortés. Semantics for Possibilsitic Disjuntive Programs. *Theory and Practice of Logic Programming*, doi: 10.1017/S1471068411000408, 2011.

[11] Juan Carlos Nieves, Mauricio Osorio, and Claudia Zepeda. A Schema for Generating Relevant Logic Programming Semantics and its Applications in Argumentation Theory. *Fundamenta Informaticae*, 106(2-4):295–319, 2011.

[12] Mauricio Osorio and Juan Carlos Nieves. Pstable semantics for possibilistic logic programs. In *MICAI 2007: Advances in Artificial Intelligence, 6th Mexican International Conference on Artificial Intelligence*, number 4827 in LNAI, pages 294–304. Springer-Verlag, 2007.

[13] Henry Prakken and Gerard A. W. Vreeswijk. Logics for defeasible argumentation. In D. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic*, volume 4, pages 219–318. Kluwer Academic Publishers, Dordrecht/Boston/London, second edition, 2002.

The 27th annual workshop of the Swedish Artificial Intelligence Society (SAIS),
14–15 May 2012, Örebro, Sweden
Published by Linköping University Electronic Press
http://www.ep.liu.se/ecp_home/index.en.aspx?issue=071

The 3 CDSSs: An Overview and Application in Case-Based Reasoning

Mobyen Uddin Ahmed, Shahina Begum and Peter Funk

School of innovation, design and engineering
Mälardalen University, SE-72123 Västerås, Sweden
mobyen.ahmed@mdh.se

Abstract* - A computer-aided Clinical Decision Support System (CDSS) for diagnosis and treatment often plays a vital role and brings essential benefits for clinicians. Such a CDSS could function as an expert for a less experienced clinician or as a second option/opinion of an experienced clinician to their decision making task. This paper presents 3 clinical Decision Support Systems as an overview of case-based reasoning (CBR) research and development. Two medical domains are used here for the case study 1) CDSS for stress diagnosis 2) CDSS for stress treatment and 3) CDSS for post-operative pain treatment. The observation shows the current developments, future direction and pros and cons of the CBR approach. Moreover, the paper shares the experiences of developing 3CDSS in medical domain.

I. INTRODUCTION

Decision Support System (DSS) that bear similarities with human reasoning have benefits and are often easily accepted by physicians in the medical domain [6, 14, 31, 32, and 33]. Hence, DSSs that are able to reason and explain in an acceptable and understandable style are more and more in demand and will play an increasing role in tomorrow's health care. Today many clinical DSSs are developed to be multi-purposed and often combine more than one AI method and technique. Many of the early AI systems attempted to apply pure Rule-Based Reasoning (RBR) as 'reasoning by logic in AI' for decision support in the medical area. However, for broad and complex domains where knowledge cannot be represented by rules (i.e. IF-THEN), this pure rule-based system encounters several problems. Knowledge acquisition bottleneck is one of the most critical problems since medical knowledge evolves rapidly, updating large rule based systems and proving their consistency is expensive. A risk is that medical rule-based systems become brittle and unreliable. One faulty rule may affect the whole system's performance and is also important to consider [8, 40]. Artificial Neural Networks (ANN) can be used in the medical domain as "reasoning by learning in AI". However, it requires large data sets to learn the functional relationship between input and output space. Moreover, transparency is another issue since the ANN functions as a so called black box i.e. it is very difficult to understand clearly what is going on [40]. Case-Based Reasoning (CBR) is a promising AI method that can be applied as "reasoning by experience in AI" for implementing DSSs in the medical domain since it learns from experience in order solve a current situation [16]. CBR is especially suitable for domains with a weak domain theory, i.e. when the domain is difficult to formalize and is empirical. In CBR, experiences in the form of cases are used to represent knowledge. A case is defined by Kolodner and Leake as "a contextualized piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoner" [26]. In practice, clinicians often reason with cases by referring and comparing previous cases (i.e. experiences). This makes a CBR approach intuitive for clinicians. A case may be a patient record structured by symptoms, diagnosis, treatment and outcome. Some applications have explored integration of CBR and RBR, e.g. in systems like CASEY [27] and FLORENCE [7]. Moreover, some of the recent medical CBR systems are studied (based on literature review) along with a survey (e-mail questionnaire to the corresponding authors) between the year 2004 and 2009 in [43]. Here, the paper investigated the current trends and developments of CBR system in medical domain.

In this paper, a case study on three CDSSs 1) CDSS for stress diagnosis 2) CDSS for stress treatment and 3) CDSS for post-operative pain treatment is presented. Here, we have discussed the power of CBR in medical domain and also presented the advancement of CBR. Moreover, our experience in order to develop 3 CDSSs on two different medical domains are presented and discussed.

II. RELATED WORK

The design and development of Decision Support Systems (DSSs) or intelligent systems in medicine is very challenging and complex. Even though the area is evolving day-by-day they are most often limited to research level. CDSSs using AI started in the early 1970s and produced a number of experimental systems; the MYCIN [8] was one of them. The HELP [19] system is one of the longest running and most successful clinical information systems. According to a literature study presented in [41], different AI techniques have been applied in the clinical DSSs such as 1) rule-based reasoning [3, 4 and 8], 2) bayesian theory [9], 3) bayesian belief networks [30], 4) heuristic, 5) semantic network, 6) neural networks [10], 7) genetic algorithms [35] 7) fuzzy logic [3, 9] and 8) case-based reasoning. Some of the recent medical DSSs using CBR approach are presented below: a) ExpressionCBR [17], the system is a decision support system for cancer diagnosis and classification. It uses Exon array data and classifies Leukemia patients automatically to help in the diagnosis of various cancer patients. b) GerAmi [14] 'Geriatric Ambient Intelligence', is an intelligent system that aims to support healthcare facilities for the elderly, Alzheimer's patients and people with other disabilities. This system mainly works as a multi-agent system and includes a CBR system to provide a case-based planning mechanism. c) geneCBR [18, 20], is focusing on classification of cancer, based on a gene expression profile of microarray data. The system is aiming to deal with a common problem in bioinformatics i.e. to keep the original set of features as small as possible. d) ISOR [38], the system identifies the causes of ineffective therapies and advises better recommendations to avoid inefficacy to support in long-term therapies in the endocrine domain. The system is exemplified in diagnosis and therapy recommendations of Hypothyroidism patients treated with hormonal therapy. e) the KASIMIR project [15], is an effort to provide decision support for breast cancer treatment based on a protocol in Oncology. The adaptation of protocol is an important issue handled here to provide therapeutic decisions for cases those are out of the protocol.

III. APPLICATION DOMAINS

In order to develop such CDSSs it requires clinical knowledge of the application domain. Medical, biological and/or physical background of a particular disease and its treatment is one example. In this paper, two different medical application domains have been discussed and presented, they are: 1) stress management and 2) post-operative pain treatment.

A. Stress management

According to Hans Selye, stress can be defined as "the rate of wear and tear within the body" [39]. We have an inborn reaction to stressful situations called the "fight or flight" response. That means we can react to certain events or facts that may produce stress and our body's nervous system activates and then stress hormones are released to protect ourselves. The wear and tear is a physiological reaction such as rise in blood pressure, rise in heart rate, increased respiration rate and muscles get ready for action.

The diagnosis of stress is often multi-factorial, complex and uncertain due to large variations and personalisation. According to [34], there are three methods that can be used for the diagnosis of stress: questionnaires, biochemical measures and physiological measures. A face-to-face interview with questionnaires and a checklist are traditional ways to diagnose stress. Rudolf E. Noble in [34], mentioned various biochemical parameters e.g. corticosteroid hormones which can be measured from body fluids, blood, saliva and urine. Since the autonomic nervous system is activated by a stress response various physiological parameters of the SNS can be used in the diagnosis of stress. The physiological parameters are commonly measured using skin conductance, Finger temperature (FT), respiration e.g. end-tidal carbon dioxide (ETCO2), electromyography (EMG), electrocardiography (ECG), heart rate e.g. calculating respiratory sinus arrhythmia (RSA) and heart rate variability (HRV), electroencephalography (EEG), brain imaging techniques, oculomotor and pupilometric measures etc.

There are several methods to control or manage stress e.g. exercise or training. In our everyday lives we need to control our stress in many situations, for instance when we are sitting at our desk or behind the wheel of a car getting stuck in traffic. In such situations or in other environments biofeedback training is an effective method for controlling stress. It is an area of growing interest in medicine and psychology and it has proven to be very efficient for a number of physical, psychological and psycho-physical problems [2, 29]. The basis of biofeedback therapy is to support a patient in realising their self-ability to control specific psychophysiological processes [25]. There is a correlation between skin temperature and relaxation. The changes in skin temperature reflect the state of the peripheral blood vessels which in turn are controlled by the SNS. A biological significant decrease in the SNS i.e. relaxation activity results in an increased diameter in the peripheral blood vessels.

In this research, both stress diagnosis and biofeedback treatment have been conducted using the skin temperature i.e. finger temperature (FT) since the intention of the research was to design and develop a CDSS for stress management which should be simple, inexpensive and easy to use.

B. Post-operative pain treatment

Approximately 40 million patients are undergoing minor to major surgical operations every year in Europe. At least half of these patients from children to elderly have suffered with a moderate or severe amount of post-operative pain. The degree of post-operative pain differs for various patients, operation site and the type of operation. For example, an operation on the thorax and upper abdomen is more painful than the lower abdomen [11]. There are different types of operations but in this project we will only focus on the following operations: 1. Cholecystectomy, 2. Total knee arthroplasty, 3. Knee arthroscopy, 4. Lower limb amputation and 5. Sternotomy for valve replacement or CABG. According to Hawthorn and Redmond [22], pain might often be a useful thing, a "protective mechanism", a biological signal, which is essential when we for example learn not to touch a stove in order to protect us from being injured. However, pain can also be a bad thing; pain after surgery obstructs the healing-process for example resistance to mobility, loss of sleep, decreased food intake, depression and loss of morale can be a consequence of post-operative pain among many other negative consequences that might occur.

The measurement of pain is very subjective and multidimensional experience and unique to every individual [12, 28]. For example, someone may experience heavy pain after a small operation and need extra medication since they have very low capacity to cope with pain. On the other hand, others may have better capacity for pain tolerance and be happy with small doses of medication. Post-operative pain has different levels and ranges starting from a minor pain to a very major acute pain. There are different ways to measure pain even if it is very subjective and individual. For example, for adults a Numerical Rating Scale (NRS) [21] or a Visual Analog Scale (VAS) [23] or Brief Pain Inventory (BPI) [13] is used and for children and elderly patients a Facial expression [24] approach can be used.

IV. THE 3 CDSSs

Clinicians/doctors have experience which may have been collected over many years. As an example, when a less experienced clinician is confronted with a new problem (for example symptoms that are not familiar) the clinician might start to analyse the whole situation and try to make a diagnosis by using their education and experience (with some solved cases). This may be a very time consuming task and may result in not finding a proper diagnosis. In that case the clinician needs to find other sources for help and a very common way is to ask senior colleagues who have more experience. A professional (more experienced clinician) might start to think to himself: "Have I ever faced any similar problem and in that case, what was that solution?" and refer the problem with their past solution to the less-experienced clinician. The less-experienced clinician then solves the problem and learns the new experience and saves it in their memory for future use. Thus, a clinical experience can be shared and reused to make a quick and correct diagnosis in the domain of health care.

Here in the 3 CDSS, a nearest neighbor (NN) algorithm is applied as a global similarity function to retrieve similar cases in CBR. The similarity measurement is taken to assess the degree of matching and creates a ranked list containing the most similar cases retrieved according to equation 1.

$$Similarity\ (C, S) = \sum_{f=1}^{n} w_f * sim(C_f, S_f) \qquad (1)$$

Where C is a current/target case, S is a stored case in the case base, w is a normalized weight defined by human expert, n is the number of the attributes/features in each case, f is the index for an individual attribute/feature and sim (C_f, S_f) is the local similarity function for attribute f in cases C and S.

A. *CDSS for stress diagnosis*

The system consists of a thermistor, sensing the finger temperature. A calibration phase helps to establish an individual stress profile and is used as a standard protocol in the clinical environment in order to collect the measurements. The protocol comprises different conditions in 6 steps, they are as follows: *baseline*, *deep breath*, *verbal stress*, *relax*, *math stress*, and *relax*. The details information about the calibration phase with the six steps can found in [5]. The steps in diagnosis stress using FT measurements and CBR is illustrated in figure 1.

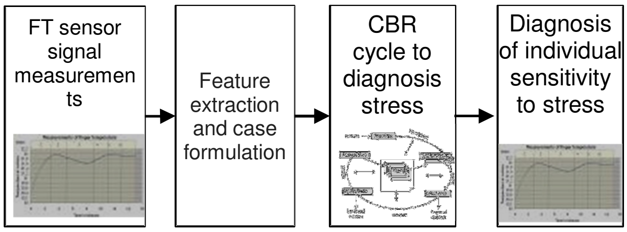

Fig 1. Steps in stress diagnosis using only FT measurements.

To determine important features the system uses 15 minute measurements (time, temperature) in 3600 samples, together with other numeric (i.e. age, room-temperature, hours since last meal, etc.) and symbolic measurements (i.e. gender, food and drink taken, sleep at night, etc.) parameters. According to closer discussion with clinicians, the derivative of each step of the FT measurement (from calibration phase) is used to introduce a "degree of changes" as an indication of the FT changes. A low angle value, e.g. zero or close to zero indicates no change or a stable finger temperature. A high positive angle value indicates a rising FT, while a negative angle, e.g. -20° indicates falling FT. The total signal, except the baseline, is divided into 12 parts each with a one minute time interval. A case is formulated with 19 features in total in which 17 features are extracted from the sensor signal (i.e. Step2_Part1, Step2_Part2, Step3_Part1, …, Step6_Part1, Step6_Part2, start temperature, end temperature, minimum temperature, maximum temperature and difference between ceiling and floor) and 2 are the human defined features (i.e. sex, hours since last meal). This new formulated case is then applied into a CBR cycle of the CDSS to assist with the diagnosis of stress.

The new problem case is then passed into the CBR cycle to retrieve the most similar cases. The case (feature vector extracted for FT signal) in this system is matched using three different local similarity algorithms [45][6]. A modified distance function uses Euclidean distance to calculate the distance between the features of two cases. Hence, all the symbolic features are converted into numeric values before calculating the distance for example, for a feature 'gender' male is converted to one (1) and female is two (2). The function similarity matrix is represented as a table where the similarity value between two features is determined by a domain expert. For example, similarity in same gender (i.e. if both are male or female) is defined by 1 otherwise 0.5. In fuzzy similarity, a triangular membership function (*mf*) replaces a crisp value of the features for new and old cases with a membership grade of 1. In both the cases, the width of the membership function is fuzzified by 50% in each side. Fuzzy intersection is employed between the two fuzzy sets to get a new fuzzy set which represents the overlapping area between them.

$$sim(C_f, S_f) = s_f(m1, m2) = \max(om/m1, om/m2) \qquad (1)$$

Similarity between the old case (S_f) and the new case (C_f) is now calculated using equation 1 where *m1*, *m2* and *om* is the area of each fuzzy set. For the interested reader, an elaborated description of fuzzy similarity can be found through the research contributions in [6, 45].

Unlike measurement-based experience, human perceptions are usually expressed in an informal and natural language format, and they are provided important information for diagnosis. In fact, when diagnosing an individual's stress level, clinicians also consider other factors such as the patient's feelings, behaviour, social factors, working environment and lifestyle. Such information can be presented by a patient using a natural text format and a visual analogue scale. Thus, the textual data of patients capture important indications not contained in measurements and also provide useful supplementary information. Therefore, the system adds textual features in a case vector which helps to better interpret and understand the sensor readings and transfer valuable experience between clinicians [45]. To enable similarity matching on less structured cases containing text, this research contributes with a proposal which combines cosine similarity with synonyms and ontology. In [45], presents a hybrid model that considers textual information besides FT sensor signal readings. For textual cases, the *tf-idf* (term frequency–inverse document frequency) [37] weighting scheme is used in a vector space model [36] together with cosine similarity to determine the similarity between two cases. Additional domain information that often improves results, i.e. a list of words and their synonyms or a dictionary provides comparable words and relationship within the words using classes and subclasses are also included. It uses domain specific ontology that represents specific knowledge, i.e. the relationship between words.

B. CDSS for stress treatment

The basis of a biofeedback system is that a patient gets feedback in a clear way (a patient observes a graph and knows from prior education how it should change). The feedback can behaviourally train the body and mind in a better way with a biological response. After discussions with clinicians it can be seen that most of the sensor-based biofeedback applications comprise of three phases illustrated in Fig. 2, 1) analyse and classify a patient and make a risk assessment, 2) determine individual levels and parameters, and finally 3) adapt and start the biofeedback training. If the clinician only uses sensor readings shown on a screen then the classification is highly experience-based [46].

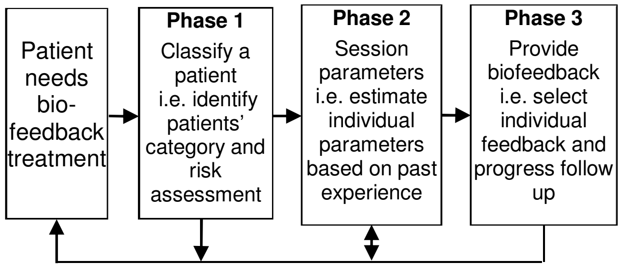

Fig 2. General architecture of a three-phase biofeedback system.

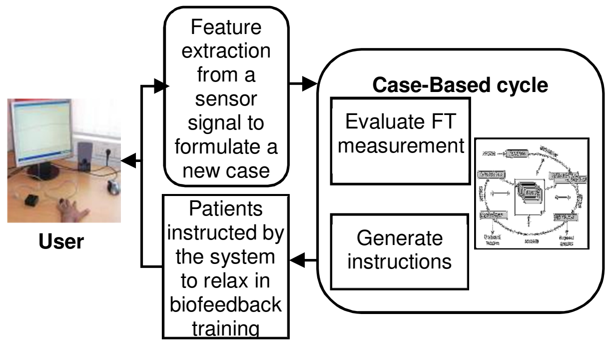

Fig 3. The steps in the biofeedback treatment cycle.

In this cycle shown in Fig. 3, a user connects a sensor to their finger and can see the changes of FT during several instructions in relaxation training. The FT measurements are performed in real time and every 2 minutes the system evaluates the last 2 minutes measurement and if necessary generates instructions for the patient. A CBR cycle is applied for the biofeedback training in stress management; this training time is flexible, which means a patient can choose the duration of their training between 6 minutes (as minimum) to 20 minutes (as maximum). Nevertheless, the system generates feedback with appropriate suggestions after every 2 minutes if necessary. Thus, for each individual, the biofeedback cases are formulated with a feature vector from a biomedical signal (i.e. with 2 minutes FT measurement) in the conditional part and a suggestion for relaxation in the solution part. A new biofeedback case is compared to previously solved cases applying a fuzzy similarity matching algorithm and displays the outcome as feedback. Here, the feedback is defined with a pair i.e. it presents an evaluation of the FT measurement and a recommendation for the next training. This generated feedback is then presented to a clinician as a proposed solution. The clinician thereafter reviews the proposed cases and takes a final decision to suggest a treatment to the patient. Thus the system assists a clinician, as a second option, to improve the patient's physical and psychological condition [46].

C. CDSS for post-operative pain treatment

A case-based system mainly depends on cases, their types and how they should be represented. The case comprises unique features to describe a problem. Cases can be presented in different ways; in the post-operative pain treatment application domain the case structure contains three parts: 1) problem, 2) solution and 3) outcome, which is slightly different from stress management domain. The cases could be defined differently on the basis of their use, manner and nature. Different DSSs might have different requirements on which type of cases are to be handled. For this domain we have proposed different types of cases namely regular cases and rare cases which is a more user-friendly format for physicians. Further, these cases are also authorised and tagged by the case owner. A short description of the different types of cases used in the system is presented in [48]. As the cases in this domain are formulated in three parts, the 'problem description' part contains around 278 attributes, and 'treatment' as a solution consists of 685 attributes, while 'outcome' as a recovery measure has 19 attributes. However, to formulate a case, feature abstraction has been done only considering the problem description and outcome information, which has been further mapped with the solution. So, out of 278 attributes only 15 features are extructed in the *problem* part and only 1 from 19 attributes were extracted from the *outcome* part. Deatailed information about feature abastruction is presented in [48]. Note that, the solution part of the cases remains unchanged since this data contains important medicine information which might modify during abstraction. Two cases are compared using nearest neighbor (NN) as global similarity and different local similarity algorithms including *modified distance function*; *similarity matrix* and *fuzzy similarity matching* discussed earlier. Only difference in the local weight which is defined by the case author(s) or owner(s) for each stored case. The weight is assumed to be a quantity reflecting the importance of the corresponding feature individually. The reason to use individual case weighting is to combine several clinicians and experts knowledge into the system.

V.OBSERVATIONS AND DISCUSSION

The paper mainly focuses on the research of the investigation of methods and techniques in order to design and develop Clinical Decision Support Systems (CDSSs). Several Artificial Intelligence (AI) methods, techniques and approaches have been investigated and applied to develop the CDSSs. Although the CBR approach is applied as a core technique for both of the domains, other AI methods were also combined and applied with CBR. In the following section of the paper presented the several related issues and a summary of the reason behind the applied methods, techniques and approaches.

A. CBR Approach Applied as a Core Technique

When the domain knowledge of any medical application is complex or not well defined such as stress management or post-operative pain treatment, it is very difficult to build a CDSS. Many of early CDSSs attempted to apply pure rule-based reasoning (i.e. IF-THEN rules), for example MYCIN [8] uses a knowledge base of ~600 rules. In order to define these 600 rules, one need to very detailed knowledge about the domain. The domain theory should be strong and well defined and this is complex and time consuming [40]. In the stress management domain, the knowledge is not well defined and there is large variation within and between patients. There is no general straight forward rule for diagnosis and treatment of stress and sometimes it is very difficult even for an experienced clinician [45][46][6]. In these situations the CBR approach works well as a CDSS since it provides the clinician with past similar cases to help them make more informed solutions. According to our previous experience in [3] where a DSS was built in order to Duodopa dose tuning for Parkinson patient using the rules defined by the domain expert. During evaluation after building the whole system, it was observed that the system can only work well on on-going (stabilized) patients but bad for new patients since these new patients don't follow any rules. Also, the same problem has been announced in post-operative pain treatment that around 30% of whole population does not fit the standard protocol [48]. In our projects using CBR the systems can learn by acquiring new cases which can be done without modification to the system [40].

The CBR approach is much better compare to Neural Network (NN) when data source is multi-media, i.e. not purely a numerical data format rather a mixture of symbolic, textual and numeric data formats. At the same time, NN requires a large data set since it divides the whole data set into three parts, training, validation and testing. Whereas a CBR system can use its whole data set in order to build and evaluate. Thus, a CBR system can start its functionality with a few cases. In the stress management domain, the NN was not used due to too few cases (i.e. only 68) and the data format is not purely numeric. But the research could apply NN in the post-operative pain treatment domain since it has more than 1500 complete cases. However, a major disadvantage of NN is that it functions as a 'black box'. The output from an NN is a utility of the weighted vectors of its neurons [40]. It cannot give any explanation or justification about the output. Moreover, it is very difficult for clinician's to trust the system in this domain as clinicians are not likely to accept any solution without an explanation. In the CBR approach the most similar cases are retrieved and presented to the clinicians to enable them to make a more informed decision. This was one of the key reasons for the clinicians in the Pain-Out project to accept the CBR approach and this is why this research has applied the CBR approach for post-operative pain treatment [48].

The research could use other techniques such as machine learning or statistics but again they require a large volume of data. The data set should be well-understood, the knowledge/hypothesis should be well-defined, there should be generalizable rules and they should be actable by rule-trace [40]. The research applied CBR as the core technique for both the application domains since CBR has process similarity i.e. it is inspired by human reasoning [1, 40]. The reasoning process is also medically accepted since doctors quite often used previous experience to solve a problem. Using CBR, an expert can directly apply their knowledge by choosing related features and their importance, for example a patient's weight is more important than the shoe size. Moreover, they can also let the system know about the similarity of two features, considering gender as a feature, a male and female may differ in similarity in e.g. choice of medication. Maintenance of a CBR system is much simpler since new knowledge can be inserted by adding new cases and the cases in case library can be used by trainee clinicians. Another interesting observation is that the CBR approach is compatible with other AI methods and thus the system can take advantage of other techniques in order to improve its performance [44] [47].

Nevertheless, some major problems that we have encountered while implementing the case-based systems are *1) Number of cases in the case library*: In CBR, the case library should contain qualitative and representative cases. The problem that we have faced in the stress domain is that the number of cases especially for a particular class 'Very Relaxed' was not enough in the case library and this has reduced the performance of the system for diagnosing new patients of that particular class. However, we have taken advantage of fuzzy rule-based reasoning to generate artificial cases which is discussed in [47] [44]. *2) Automatic adaptation*: automatic adaptation often requires clear domain knowledge. Therefore, in medical CBR systems, there is not much adaptation. The DSSs described here do not contain any automatic adaptation since in such systems where the best cases are proposed to the clinician as suggestions of solutions and when the domain knowledge is not clear enough automatic adaptation is difficult to develop or not advised [40]. However, in the presented CDSSs the solution of a past case often requires adaptation to find a suitable solution for the new case. This adaptation might often be a combination of two or more solutions of cases from the retrieved cases. Therefore, retrieving a single matching case as a proposed solution may not be sufficient for the DSSs. So, the systems retrieve a list of ranked cases to help in the adaptation process but finally the adaptation and validation of cases has been carried out by clinicians in the domains.

B. Others AI Techniques Applied as Tools

Besides the CBR approach, the research has investigated and applied several AI technologies and approaches. The other AI technologies have been used as tools help to fine tune the CBR approach in order to obtain the complete benefit of the CBR systems. The combination of several AI techniques depends on the nature of the domain, data format, complexity and performance. For both the application domains, fuzzy logic is applied in the similarity measurement of the CBR approach since it helps to accommodate uncertainty. Moreover, fuzzy similarity matching presented in [44][45][46] reduces the sharp distinction which is obtained in similarity matrix [6] formulated by an expert. In [44] section 3.1, comparisons of three local similarities are presented and it shows that when using fuzzy similarity the system can achieve a better performance. The information retrieval approach i.e. mainly vector space modeling (VSM) is applied together with a WordNet dictionary and domain specific ontology in order to retrieve textual cases presented in [45]. Here, the system did not combine textual cases and FT cases in order to provide a combined solution rather the most similar textual cases are presented. These textual cases provide information regarding patients' contextual feelings, behaviors, social facts, working environments, lifestyle and other additional information which enhances the reliability of the CDSS. Thus, the CDSS for stress management provides support to clinicians in their decision making tasks by only retrieving the most similar cases (both textual cases and FT measurements cases simultaneously) rather than providing any combined direct solution.

To some extent, when the domain knowledge is well-defined the RBR can work, for example, when FT increases the patient is in a relaxed state and when FT is decreasing it is usually a sign of a stressed state. These simple rules can be used and in this research they are used only to create artificial cases [47]. So, in the stress management domain, fuzzy rule-based reasoning is used together with a CBR approach in order to improve the system's performance since there are very few reference cases in the case library. In [44] section 3.2, the result shows that the CDSS succeed to improve its performance by 22% by using a bigger case library initiated both by using reference cases and cases created by fuzzy rules. The details of the fuzzy rule-based reasoning scheme and related experimental works are presented in [47].

C. Future Direction and Application

CBR system can provide a case library which motivates a new direction of data mining and knowledge discovery and combination with CBR. For example in post-operative pain treatment domain, there were more than 1500 reference cases in the case library, so the clustering approach in terms of data mining and knowledge discovery could be used in order to identify unusual cases (outliers) from the regular ones[49]. Here, a novel combination of the FCM and Hierarchical clustering which was able to identify 18% of cases as rare cases. This group of unusual case together with regular one can be stored into the case library and the retrieval function of the CBR system could consider the most similar groups and present to a clinician accordingly. This could also help to reduce retrieval time in the CBR system.

Currently, the stress diagnosis system is expanded to use it in a professional driving situation [42]. The driver cases could lead to new research challenges in developing the CBR systems and meeting these challenges could positively impact on the advancement of CBR systems for health care in professional environment.

Acknowledgments. The authors gratefully acknowledge the project PainOut funded by EU FP-7, the European Commission (7th Framework Programme, Call HEALTH-2007-3.1-4, IPOS, Integrated Personal Health Optimizing System funded by KKs, NovaMedTech EU objective 2 project and PROEK, Ökad Produktivitet och Livskvalitet funded by Sparbanksstiftelsen for their support of these research projects and also like to thank the participating patients and clinicians for their contributions to this work.

REFERENCES

[1] Aamodt A. and Plaza E., Case-based reasoning: Foundational issues, methodological variations, and system approaches, AICom - Artificial Intelligence Communications, IOS Press, Volume 7, Issue 1, pp 39-59, 1994.

[2] AAPB, The association for applied psychophysiology and biofeedback, http://www.aapb.org/, Last referred November 2011.

[3] Ahmed M. U., Westin J., Nyholm D., Dougherty M. and Groth T., A fuzzy rule-based decision support system for duodopa treatment in Parkinson, Proceedings of the 23rd annual workshop of the Swedish Artificial Intelligence Society, Umeå, May 10-12, Ed(s): Eklund P., Minock M. and Lindgren H., pp 45-50, 2006.

[4] Alayón S., Robertson R., Warfield S.K., and Ruiz-Alzola J., A fuzzy system for helping medical diagnosis of malformations of cortical development, Biomedical Informatics, Volume 40, Issue 3, pp 221-235, 2007.

[5] Begum S., Ahmed M.U., Funk P., Xiong N. and Scheele B. V., Using calibration and fuzzification of cases for improved diagnosis and treatment of stress, The Proceedings of the 8th European Workshop on Case-based Reasoning, pp 113-122, 2006.

[6] Begum S., Ahmed M.U., Funk P., Xiong N. and Schéele B.V, A case-based decision support system for individual stress diagnosis using fuzzy similarity matching, Computational Intelligence, Blackwell Publishing, Volume 25, Nr 3, pp 180-195, 2009.

[7] Bradburn C. and Zeleznikow J., The application of case-based reasoning to the tasks of health care planning. In proceedings of topics in case-based reasoning: 1st European workshop, EWCBR-93, Springer, Berlin, pp 365–378, 1994.

[8] Buchanan B.G. and Shortliffe E.H., Rule-based expert systems: The MYCIN experiments of the Stanford heuristic programming project, Reading Mass. : Addison-Wesley, 1984.

[9] Casas F., Orozco A., Smith W. A., De Abreu-García J. A. and Durkin J., A fuzzy system cardio pulmonary bypass rotary blood pump controller, Expert Systems with Applications, Volume 26, Issue 3, pp 357-361, 2004.

[10] Charles E. K, Linda M. R., Katherine A. S. and Peter H., Construction of a Bayesian network for mammographic diagnosis of breast cancer, Computers in Biology and Medicine, Volume 27, Issue 1, pp 19-29 January 1997.

[11] Charlton E., The management of postoperative pain, update in anaesthesia, World Federation of Societies of Anaesthesiologists, Issue 7, Ar. 2, 1997, http://www.nda.ox.ac.uk/wfsa/html/u07/u07_003.htm, Last referred November 2011.

[12] Clark W.C., Yang J.C., Tsui S.L., Ng K.F. and Clark S.B., Unidimensional pain rating scales: a multidimensional affect and pain survey (MAPS) analysis of what they really measure, Pain, Volume 98, Issue 3, pp 241-247, 2002.

[13] Cleeland C.S. and Ryan K.M., Pain assessment: Global use of the brief pain inventory, Annals of the Academy of Medicine Singapore, Volume 23, Issue 2, pp 129–138, 1994.

[14] Corchado J.M., Bajo J. and Abraham A., GERAmI: Improving healthcare delivery in geriatric residences, IEEE Intelligent Systems, Volume 23, Issue 2, pp 19-25. ISSN: 1541-1672, 2008.

[15] D'Aquin M., Lieber J. and Napoli A., Adaptation knowledge acquisition: A case study for case-based decision support in oncology, Computational Intelligence, Volume 22, Issue 3-4, pp 161 – 176, 2006.

[16] Demirali J., Reason: An introduction to critical thinking, 1st edition. ISSBN 978-1-4583-7749-4, Analogical Reasoning Institute, 2011.

[17] De Paz J.F., Rodriguez S., Bajo J. and Corchao J.M., Case-based reasoning as a decision support system for cancer diagnosis: A case study, International Journal of Hybrid Intelligent Systems (IJHIS), IOS press, Volume 6, Issue 2, pp 97-110, 2009.

[18] Díaz F., Fdez-Riverola F. and Corchado J.M., GENE-CBR: A case-based reasoning tool for cancer diagnosis using microarray data sets, Computational Intelligence, Volume 22, Issue 3-4, pp 254-268, 2006.

[19] Gardner R.M., Pryor T.A. and Warner H.R., The HELP hospital information system: update 1998, International Journal of Medical Informatics, Volume 54, Issue 3, pp 169-182, 1999.

[20] Glez-Peña D., Díaz F., Hernández J.M., Corchado J.M. and Fdez-Riverola F., geneCBR:A translational tool for multiple-microarray analysis and integrative information retrieval for aiding diagnosis in cancer research, BMC Bioinformatics, Volume 10, Issue 1, Publisher: BioMed Central, 2009.

[21] Hartrick C.T., Kovan J.P. and Shapiro S., The numeric rating scale for clinical pain measurement: a ratio measure?, Pain Pract, Volume 3, Issue 4, pp 310–316, 2003.

[22] Hawthorn J. and Redmond K., Pain: Causes and management, Journal of Psychiatric and Mental Health Nursing, Volume 6, Issue 5, pp 409–410, 1999.

[23] Huskisson E.C., Measurement of pain, The Lancet, Volume 304, Issue 7889, pp 1127-1131, 1974.

[24] Jose D.A., Fischer H.B.J., Ivani G., Mogensen T., Narchi P., Singelyn F.J., Stienstr R. and Wulf H., Postoperative pain management – good clinical practice, European Society of Regional Anaesthesia and Pain therapy, http://www.esraeurope.org/PostoperativePainManagement.pdf, Last referred September, 2011.

[25] Kappes B., Biofeedback therapy: Training or treatment, Applied Psychophysiology and Biofeedback, Volume 33, pp 173-179, 2008.

[26] Kolodner J.L. and Leake D., A tutorial introduction to case-based reasoning, In Case-Based Reasoning: Experiences, Lessons and Future Directions, Leale D. (Ed.), Cambridge, MA: AAAI/MIT Press, pp 31-65, 1996.

[27] Koton P., Using experience in learning and problem solving, Massachusetts Institute of Technology, Laboratory of Computer Science, PhD Thesis MIT/LCS/TR-441, 1989.

[28] Kumar S., Tandon O.P. and Mathur R., Pain measurement: a formidable task, Indian Journal of Physiol Pharmacol, Volume 46, Issue 4, pp 396–406, 2002.

[29] Lehrer P., Smetankin A. and Potapova T., Respiratory Sinus Arrhythmia Biofeedback Therapy for Asthma: A Report of 20 Unmedicated Pediatric Cases Using the Smetankin Method, Applied Psychophysiology and Biofeedback, Volume 25, Issue 3, pp 193-200, 2000.

[30] Lindgaard G., Egan P., Jones C., Pyper C., Frize M., Walker R., Boutilier C., Hui B., Narasimhan S., Folkens J. and Winogron B., Intelligent decision support in medicine: back to Bayes?, Journal of Universal Computer Science, Volume. 14, Issue 16, pp 2720- 2736, 2008.

[31] Marling C. and Whitehouse P., Case-based reasoning in the care of Alzheimer's disease patients, 4th International Conference on Case-Based Reasoning, ICCBR 01, pp 702-715, 2001.

[32] Montani S., Portinale L., Leonardi G. and Bellazzi R., Case-based retrieval to support the treatment of end stage renal failure patients, Artificial Intelligence in Medicine, Volume 37, Issue 1, pp 31-42, 2006.

[33] Nilsson M., Funk P., Olsson E., Von Schéele B. H. C., and Xiong N., Clinical decision-support for diagnosing stress-related disorders by applying psychophysiological medical knowledge to an instance-based learning system, Artificial Intelligence in Medicine, Volume 36, Issue 2, pp 159-176, 2006.

[34] Noble R.E, Diagnosis of stress, Metabolism - Clinical and Experimental, Elsevier Science (USA),Volume 51, Issue 6, Part B, pp 37-39, 2002.

[35] Podgorelec V., Kokol P. and Zavrsnik J., Medical diagnosis prediction using genetic programming, 12th IEEE Symposium on Computer-Based Medical Systems, pp 202 - 207, 1999.

[36] Salton G., Wong A. and Yang C.S., A vector space model for automatic indexing, Communications of the ACM, Volume18, Issue11, pp 613–620, 1975.

[37] Salton G. and Buckley C., Term weighting approaches in automatic text retrieval, Information Processing and Management, Volume 24, Issue 5, p513-23, 1988.

[38] Schmidt R. and Vorobieva O., Case-based reasoning investigation of therapy inefficacy, Knowledge-Based Systems, Volume 19, Issue 5, pp 333–340, 2006.

[39] Selye H., The Stress of Life. New York: McGrawHill, 1956, Revised edition 1976.

[40] Watson I., Applying Case-Based Reasoning, Techniques for Enterprise Systems, Morgan Kaufmann Publishers Inc, 250 pages, ISBN 1-55860-462-6, 1997.

[41] Yang J.B., Xu D.L. and Kong G., Clinical decision support systems: a review on knowledge representation and inference under uncertainties, International Journal of Computational Intelligence Systems, Volume 1, Nr. 2, pp 159-167, 2008.

[42] Begum S., A Case-Based Driver's State Monitoring System Using Physiological Sensor Signals, 11th Industrial Conference, ICDM 2011, Advances in Data Mining, Workshop Proceedings on Case-Based Reasoning , p 42-54, Ibai Publishing, New York, USA, Editor(s):Petra Perner and Georg Rub, September, 2011.

[43] Begum S., Ahmed M.U., Funk P., Xiong N., Folke M., Case-Based Reasoning Systems in the Health Sciences: A Survey on Recent Trends and Developments, International journal of IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, vol 41, Issue 4, 2011, pp 421 - 434.

[44] Ahmed M.U., Begum S., Funk P., A Hybrid Case-Based System in Stress Diagnosis and Treatment, Accepted in the IEEEEMBS International Conference on Biomedical and Health Informatics (BHI2012), 2012.

[45] Ahmed M.U., Begum S., Funk P., Xiong N., von Schéele B, Case-based Reasoning for Diagnosis of Stress using Enhanced Cosine and Fuzzy Similarity. International journal of Transactions on Case-Based Reasoning on Multimedia Data, vol 1, Number 1, IBaI Publishing, ISSN: 1864-9734, 2008, pp 3-19.

[46] Ahmed M.U., Begum S., Funk P., Xiong N., von Schéele B, A Multi-Module Case Based Biofeedback System for Stress Treatment. International journal of Artificial Intelligence in Medicine, vol.51, Issue 2, Publisher: Elsevier B.V., 2010, pp 107-115.

[47] Ahmed M.U., Begum S., Funk P., Xiong N., Fuzzy Rule-Based Classification to Build Initial Case Library for Case-Based Stress Diagnosis. In the proceedings of 9th International Conference on Artificial Intelligence and Applications (AIA), 2009, pp 225-230.

[48] Ahmed M.U., Funk P., A Case-Based Retrieval System for Post-operative Pain Treatment, In the proceeding of International Workshop on Case-Based Reasoning CBR 2011, IBaI, Germany, New York/USA, Ed(s):Petra Perner and Georg Rub, September, 2011, pp 30-41.

[49] Ahmed M.U., Funk P., Mining Rare Cases in Post-Operative Pain by Means of Outlier Detection, Accepted in the IEEE International Symposium on Signal Processing and Information Technology, 2011.

The 27th annual workshop of the Swedish Artificial Intelligence Society (SAIS),
14–15 May 2012, Örebro, Sweden
Published by Linköping University Electronic Press
http://www.ep.liu.se/ecp_home/index.en.aspx?issue=071

Money Laundering Detection using Synthetic Data

Edgar Alonso Lopez-Rojas
School of Computing
Blekinge Institute of Technology
edgar.lopez@bth.se

Stefan Axelsson
School of Computing
Blekinge Institute of Technology
stefan.axelsson@bth.se

Abstract

Criminals use money laundering to make the proceeds from their illegal activities look legitimate in the eyes of the rest of society. Current countermeasures taken by financial organizations are based on legal requirements and very basic statistical analysis. Machine Learning offers a number of ways to detect anomalous transactions. These methods can be based on supervised and unsupervised learning algorithms that improve the performance of detection of such criminal activity.
In this study we present an analysis of the difficulties and considerations of applying machine learning techniques to this problem. We discuss the pros and cons of using synthetic data and problems and advantages inherent in the generation of such a data set. We do this using a case study and suggest an approach based on Multi-Agent Based Simulations (MABS).

Keywords: Machine Learning, Anti-Money Laundering, Money Laundering, Anomaly Detection, Synthetic Data, Multi-Agent Based Simulation

1 Introduction

Money laundering threatens the economic and social development of countries. The threat is due to the injection of illegal proceeds into the legitimate financial system. Due to the high amount of transactions and the variety of money laundering tricks and techniques, it is difficult for the authorities to detect money laundering and prosecute the wrongdoers. Thus, it is not only the amount of transactions, but the ever changing characteristics of the methods used to launder money that are constantly being modified by the *fraudsters*, which makes this problem interesting to study.

This paper aims to analyze the implications of using machine learning techniques for money laundering detection (also known as Anti-Money Laundering, AML) in a data set consisting of synthetic financial transactions.

Our case study is based on the company **AB** [1]. Company **AB** has developed a mobile money implementation that provides users with the ability to transfer money between mobile phone users, by using the phone as a sort of electronic wallet. The task at hand is to provide a tool that detects suspicious money laundering activities.

The mobile money service is currently running in a demo phase. Hence, real data from this system is not available at this stage and therefore the system does not produce representative data that can be used e.g. for the training of the machine learning algorithm. Thus, due to the lack of real data we turn to the generation of synthetic data as an alternative.

The use of synthetic data for machine learning has implications. In this paper we present our ideas about how to address some of the difficulties raised by the lack of real data.

Outline The rest of this paper is organized as follows: sections 2 and 3 introduce the topic of money laundering and present previous work. Sections 4 and 5 address the main topic, which is the use of synthetic data for Anti-Money Laundering. We finish with a discussion and conclusions including future work in sections 6 and 7.

2 Background

Money Laundering affects the finances of nations and it may contribute to an increase in the funding

[1] The identity of the Company AB unfortunately cannot be disclosed due to a Non-Disclosure Agreement

of criminal activities [3]. Due to issues such as the high amount of transactions taking place in any financial service, it is not a trivial task to find specific anomalous transactions that should be marked as suspicious. The reported suspicious activity needs to be supported by tangible evidence that allows specialized government agencies to investigate further.

In Sweden and other countries, most companies in the financial sector are required by law to address money laundering detection. The cost of implementing such controls for AML is quite high, mainly because of the amount of manual labor required. In Sweden alone it is estimated to be around 400 million SEK annually [13].

The most common method today used for preventing anomalous financial transactions consist in establishing thresholds for all transactions. Transactions that exceed these thresholds require extra scrutiny, whereby the client needs to declare the precedence of the funds. These thresholds are set by law without distinction made between different economic sectors or actors. However, this of course leads to fraudsters changing their behavior in order to avoid this kind of control, by e.g. making many smaller transactions that fall just below the threshold [9].

The specific domain covered here is the service *Mobile Money*[2], which is offered by Company **AB**. *Mobile Money* is a platform for transferring money between users, using their mobile phones. This is accomplished by the use of codes sent through text messages or the Internet.

Mobile money brings several benefits for users, including the simplicity of transferring money between themselves and others. One user only needs to know the mobile phone number of the receiving user in order to send money. If the receiving users are registered in the system then the money can be deposited right away in their account. Otherwise, the users receive a code via SMS that enables the recipient to collect the money in cash at one of the nearby local stores that are affiliated with the mobile money operator. It does not require a user to own a bank account, which is beneficial for many people in the world who do not have sufficient assets to warrant a bank account. However, if the user wants to refill their account or withdraw money, then an existing bank account can be connected to the mobile money account, and used in conjunction with it. There are also other alternatives such as top-up card or credit cards connected with the service, that can be used to deposit or withdraw money from the mobile money system.

[2] *Mobile Money* is a generic name that we use in this study and it is not the true name of the service provided by Company **AB**

3 Related work

A number of basic countermeasures against money laundering have been proposed, including basic statistical analysis which constrains the amount of the transactions as well as restricting their frequency [5]. Other methods that complement these basic security measures are based on checking every customer against a black list originating from previous investigated cases and a white list to e.g. avoid mistakes when faced with persons with the same name. Unfortunately, these and other methods have proved to be insufficient [13].

Several machine learning techniques have been used for detecting fraud, and more specifically money laundering, [17]. From the point of view of machine learning, the application is interesting, due to the successful classification rate (high *True Positives* and low *False Positives*) that the classification model can achieve compared to other methods such as simple rule based detection that compares transactions against fixed thresholds.

Data mining based methods have also been used to detect fraud [15, 20, 19]. This leads to the observation that machine learning algorithms can identify novel methods of fraud by detecting those transactions that are different (suspicious) in comparison with the benign transactions. This problem in machine learning is known as novelty detection. Supervised learning algorithms have been used on a synthetic data set to prove the performance of outliers detection [1].

There are tools such as IDSG (IDAS Data and Scenario Generator [11]) which was developed with the purpose of generating synthetic data based on the relationship between attributes and their statistical distributions. IDSG was created to support data mining systems during their test phase and it is been used to test fraud detection systems.

Gao [8] proposed one of the frameworks used for AML introduces the terms *legal transaction, usual transaction, unusual transaction, suspicious transaction and illegal transaction* for describing different possible categories of transactions. This framework aims to rank the likelihood that a transaction would be illegal on a scale from 0 to 100, which enables prioritization.

We wish to stress that a detector cannot be completely certain that a transaction corresponds to money laundering. This task is delegated to the legal authorities. Instead of doing that, we intend to flag customers and transactions with a label of suspicion that focus the attention of the operator for further investigations.

Despite the possible bias injected in the data set during the simulation, synthetic data has been previously used with similar reasons as the ones presented by Barse [2]. As in that work, the *lack of real data* and the *low probability of real instances of fraud in the real world data obtained*, are some of the reasons discussed further in this paper.

In general the availability of financial information for research is very restricted by the corporate policies and even the law. Customers are usually protected by the financial organizations and the disclosure of their private information is limited by internal and government policies. In order to get access to such data, anonymization techniques should be used on the data set in order to allow for the preservation of privacy [18, 10].

4 AML for mobile money

The detection of money laundering in the mobile money service is not trivial due to the difficulty of classifying transactions that are intended to appear as normal and legal. In this paper we address this problem with the approach of learning from the experiences of past detected patterns of illegal behavior in order to gain more knowledge about the possible rules or new patterns of fraud that could emerge in a mobile money system.

The first step to address our problem is to start by clearly formulating the learning problem.

4.1 Problem definition

Regarding the mobile money AML domain, we formulate the learning problem as:

Task T Classification of transactions as normal or suspicious based on the known pattern of legal transactions. The aim is to find anomalies or outliers inside a data set of mobile money financial transactions.

Performance Measure P Percentage of transactions correctly classified as anomalous, also known as *True Positives* (TP) and the percentage of *False Positives* (FP) i.e. transactions that are not anomalies and are misclassified as anomalies.

Experience E Synthetic data generated with transactions labeled as legal (normal) and/or illegal (suspicious).

The main weakness is that the experience gained by using a synthetic data set can be biased and in some cases it may not match a realistic situation that would occur in a real-world data set. However in the following sections we present our analysis of how this can also be used to our advantage by allowing us to gain information about unseen but expected situations in a real-world data set.

4.2 Data Preprocessing

One of the tasks that need to be addressed is data preprocessing. This task includes the selection of the attributes, discretization, noise removal and, in certain domains, data fusion.

Company **AB** has a design with a database that aims to store all the log information about the users interactions with the service. For this study we need to select the attributes that best contribute to the correct classification of suspicious transactions.

The customers are originally differentiated in the system by a profile. The profile for each customer is specified at the opening of the account by internal criteria. Customers with certain profiles are limited e.g. in the amount and the frequency of the transactions that they can perform. Additionally, there are specific methods that detect anomalies based on profiling customers which can be applied when faced with prior profiles such as these [4].

In addition to the generation of profiles, we added the following attributes to our simulation: Customer ID, Profile, Date of the Transaction, type of transaction (e.g. deposit, withdraw, transfer), Amount of the Transaction, Location (city), Account Age (months since the creation of the account) and Customer Age (years).

For each transaction of type *transfer* there is also a *deposit* transaction with the same value for a different customer. This transfer transaction describes a social network between customers. The rest of the fields are generated according to the given parameters of the simulation and random operations with range validation to guarantee consistent data that follows a realistic model.

Data labeled as anomalous should be added in the transaction database in order to run supervised algorithms. These anomalous records are created with the intention to replicate some of the common patterns used by fraudsters. One example of these patterns is an unverified user profile which makes either large deposits or large withdraws in comparison to a predefined threshold. Some other known problematic patterns of usage include: several withdraws from the same profile above the average normal value for transactions by young customers, a verified customer that performs a single large withdraw, and finally a chain of transactions that deposits money in a single account followed by consecutive withdrawals from that same account.

In a realistic situation we would be handling millions of transactions in a data set and in most of the cases only a small testing samples of the whole data set can be processed.

Although computational scaling performance is a topic that is not addressed here, the learning algorithms selected are profoundly affected by the amount of data provided for the training and the cross validation phase.

4.3 Learning with Synthetic Data

For a real world data set the selected algorithms should produce the best accuracy, i.e. TP rate, in comparison with the other algorithms. This tells us directly that our classifier can detect a significant number of suspicious transactions. However, the FP rate which is represented by the misclassified number of instances from the normal data, is also an important indicator of performance because we do not want a situation where the high number of FP will consume the time of an investigator and leads to possible missed cases and lack of trust by the investigators in the detector.

Thus, we are interested in providing an accurate method to improve the detection rate (TP) and reduce the misclassification rate of the benign data (FP) counted on the data collected from the simulation. A synthetic data set can be used to train the classifier and test different scenarios. One example of such a scenario is when the customer population have low income and only a few customers have large assets.

The results of this classification algorithm with a synthetic data set should be interpreted in a broader perspective than results with a real-world data set. In a real scenario the results are used for prosecuting and reporting individuals. But when doing research using a synthetic data set the purpose is different. One of the goals is to identify the measures of detection and control that could be added to the system, given a set of conditions, with the clients.

We studied possible algorithms for our detection research using the same data set. The algorithms analyzed here are based on *Decision Tree* learning and clustering techniques. Other methods such as Support Vector Machine (SVM), Neural Networks, Link Analysis and Bayesian networks are not addressed here, but we expect to improve our approach in a future work by including these methods.

Decision Tree algorithms construct a tree that contains branches with rules that correctly classifies the most of the data set [16, 6]. The main advantage of using these algorithms (in comparison with other machine learning algorithms) for the domain of mobile money AML is the possibility for an investigator to determine common rules that classify suspicious behavior.

There could be situations where *fraudsters* start to behave according to a new pattern based on e.g. a specific location or city, combined with other attributes such as age, profile and others that inspected singly seem to be normal, but in combination could lead to the detection of a new fraud trend. However some of the requirements of these algorithms are that the data set used for the training should be representative enough of the whole data set in order to get rules that sufficiently generalize to the real scenario. These rules should be refreshed frequently in order to detect new possible fraud trends.

Clustering techniques such as distance based clustering and density based clusters can be useful to classify natural clusters that appear in the data set. The disadvantage with these techniques is the hard task of finding the parameters that expose abnormal behavior clusters due to the class unbalance problem of the distribution of the classes [7]. In a normal situation most of the records in a data set are instances of the normal behavior of a customer, with only a few representing anomalous (i.e. interesting) behavior.

In addition, the complexity of the patterns used by *fraudsters* represent a challenge, due to the fact that the fraudsters' patterns intend to mimic normal behavior in order to pass undetected by law enforcement.

Besides, from the perspective of our detection tool, we cannot be 100% certain of the illegal precedence of the funds in a transaction, that is why our detector should include a *suspicious* rank that allows an investigator to prioritize more relevant and important cases.

5 Mobile Money simulation

During this research, we found a number of difficulties that affect working with this domain. We found the lack of access to real data, the poor quality of samples in the real-world data we could access, and the many possible scenarios that we would like to explore, are some of the problems found in our case study. Such a problems make the process of research more difficult. This is the reason behind the discussion of using synthetic data as an alternative to further research in this area.

Important issues arise when analyzing the use of machine learning for money laundering such as: volume and complexity of data, class imbalance, concept drift, class overlap and class mislabeling [17].

However, many other considerations also play a role in the simulation. One of the most important challenges that we need to address is: *How can we make our model realistic and as close as possible to a desired scenario?*

In order to answer this question, we present the following benefits and disadvantages of using synthetic data for our research.

5.1 Benefits of using synthetic data

When using synthetic data one of the benefits we identified is the possibility of selecting attributes that reduce considerably the complexity of the data structures involved. Furthermore, this simplifies the tasks of data preparation and extraction from real sources. The volume of the data can be tuned to comply with different experimental setups.

The class imbalance problem can be reduced by setting up a simulation that produces enough records of each of the interesting classification classes. Class overlap is still a remaining issue with simulations. However, simulations that properly represent fraud behavior can avoid class mislabeling.

We summarize our findings as:

- The data that represent realistic scenarios are readily available.
- The privacy of the customer is not impacted.
- The disclosure of results is not affected by policies or legal issues.
- The data set is available for other researchers to reproduce experiments.
- Different scenarios can be modeled with parameters controlled by the researcher.
- Injection of enough abnormal data to address the class unbalance problem.
- Simulation of abnormal behavior prevent the problem of mislabeled classes.

5.2 Disadvantages of using synthetic data

Unfortunately other issues arise that are important to consider when using synthetic data. Some of these issues are:

- The data generated might be nor representative or realistic.
- Data can have biased information.
- It is difficult to build a realistic model due to the complexity of variables and parameters.
- The simulated suspicious data cannot be investigated further by the government agency. In a real scenario these results could be used for improving the accuracy of the existing classification algorithms.
- It is unknown if we can transfer the learning from a simulated data set to a real-world data set.

Some of the disadvantages presented can be minimized if we can build a simulation with records that can represent a real-world situation. It is important to understand that the purpose of the simulation is not to reproduce a view of the real world, but to provide an alternative simplified scenario that is designed according to a model as it is presented in the following section.

5.3 Multi-Agent Based Simulation

Multi-Agent Based Simulation is an approach that involves the use of autonomous and interactive agents and it has been used to model complex systems. These agents are described by their state, behavior and their interaction with other agents, which generates complex global behavior usually found in different domains [12].

Previous work has shown the use of Multi-Agent based simulation in the task of simulating social networks and analyzing social behavior [14]. *Mobile Money* resembles a social network of connected clients where the connections are represented by the transactions (money sent or received) and the nodes are represented by the clients.

The synthetic data from a simulation aims to represent the interactions of the customers of a mobile money system. The graph shown in Figure 1 is used to represent a desired scenario that can be used to study a certain phenomenon nor existing in a real world data set. This graph is an hypothetical situation where 2000 clients from 7 different cities perform legal transactions with customers inside or outside their cities. The simulation allows the researcher to follow one agent and keep track of its behavior and also store all the transactions for further analysis.

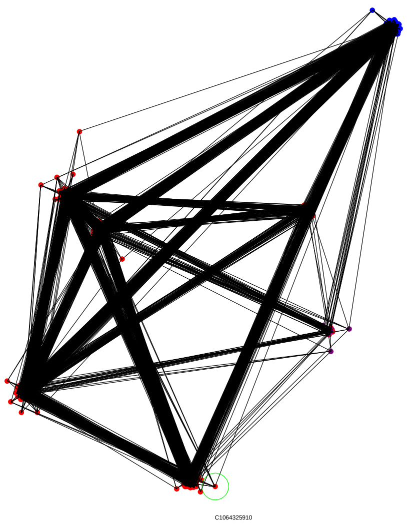

Figure 1: Scenario 1 - 2000 clients distributed across 7 cities and multiple edges connecting clients that produces legal transactions

Figure 2 represent a small simulation of 20 malicious agents distributed across 3 cities. The behavior of these agents can be modeled as a cooperative network of agents which aim to move a certain amount of money from the red nodes to the blue nodes. By doing this, the malicious agents avoid the threshold controls present in the system.

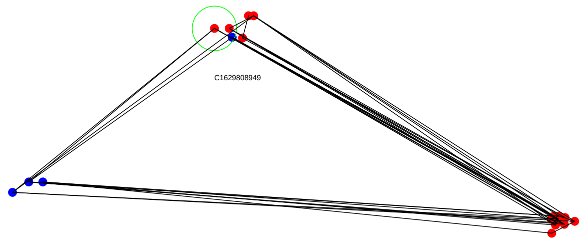

Figure 2: Scenario 2 - 20 accounts distributed across 3 cities generating suspicious transactions

There are several agent-based frameworks that incorporate toolkits to aid the development of these kind of systems. Some of them are freely available and are widely used in academic simulations (e.g. MASON[3], Repast [4], Swarm[5]). We used MASON for the network graphs and simulations presented in Figure 1 and 2.

6 Discussion

One of the difficulties with this domain is the lack of available data sets to use for training the machine learning algorithms and compare results with other researchers. This is the reason why a synthetic data set is proposed as an alternative. We are unable to obtain any data set for the mobile money system at the moment.

Before, we have done some preliminary work based on a simplified statistical simulation. Because of the elementary of this simulation we are not going to discuss any result here. This simulation was a biased representation of the domain that was used to run different machine learning algorithms in order to experiment with different outputs. But from this experience, we gained interest in producing a better simulation that shows real behavior and characteristics of clients from this domain.

The concept of MABS presented in section 5.3 make us consider that we can obtain a model based

[3]MASON `http://cs.gmu.edu/~eclab/projects/mason/`
[4]Repast `http://repast.sourceforge.net/`
[5]Swarm `http://www.swarm.org/`

on multi-agents that would be realistic enough for our purpose of analyzing the possible scenarios and build a detector for money laundering. This led us to work on a simulation based on the concept of Multi-Agents Systems.

The problem of finding anomalies within the domain of money laundering is really a challenge. Every time a new pattern of money laundering is detected by the authorities and a new control mechanism is implemented, the fraudsters change their modus operandi and create a new method that is undetectable by the current rules of a rule based AML system.

When doing research in the domain of *mobile money* a number of difficulties arise, including the lack of access to real data that can be used to evaluate the learning algorithms. Even with real data, the lack of anomalous transactions can be a problem. This is why the injection of synthetic anomalies in a real-world data set is an alternative to overcome the problem of e.g. an unbalanced number of classes.

We do not expect to be able to identify all anomalies but we intend to identify abnormal behavior from the customers that can lead to the detection of these new mutated methods.

7 Conclusions

We have presented an analysis of the use of a synthetic data set from the domain of mobile money for experimentation with machine learning algorithms. Through the use of simulation of different scenarios we can discover flaws in the current system. This can also lead to the finding of new policies and legislation that could detect the appearance of previous detected patterns of money laundering in the future.

We pretend to illustrate the methods that can be used to evaluate the accuracy of different algorithms, without going into specific details. Our analysis covers *Decision Trees*, *Clustering* techniques and *Decision Rules* that are more understandable by human operators than other machine learning algorithms.

When working with synthetic data there is always a risk of generating a data set that does not represent the real world data set. This can lead to results that are biased by the way the data was generated. On the other hand a synthetic data set can also simulate different scenarios that are not available for experimentation and analysis as they are unusual, catastrophic etc.

As shown before, the benefits presented in section 5.1 make us conclude that using synthetic data for machine learning experimentation is a good alternative in domains where the lack of real data is a problem.

Further work will focus on building a model for the simulation of mobile money transactions. Multi-Agent Based Simulation (MABS) is an interesting technique that can be used to improve the results of the generation of realistic synthetic data sets for this domain. We aim to test in the future the performance of several machine learning algorithms such as Support Vector Machine (SVM), Neural Networks, Link Analysis and Bayesian Networks. These algorithms have been used successfully in previous studies and it is of our interest to evaluate them in future research.

Acknowledgments This research is supported by the research group DISL (Distributed and Intelligent Systems Laboratory) from BTH (Blekinge Techniska Högskola) in Sweden. We also thank PhD Niklas Lavesson (BTH) for his contribution and reviews of this article.

References

[1] Naoki Abe, Bianca Zadrozny, and John Langford. Outlier detection by active learning. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*, page 504, 2006.

[2] E.L. Barse, H. Kvarnstrom, and E. Johnson. Synthesizing test data for fraud detection systems. *19th Annual Computer Security Applications Conference, 2003. Proceedings.*, pages 384–394, 2003.

[3] B. Bartlett. The negative effects of money laundering on economic development. *Asian Development Bank Regional Technical Assistance Project No*, (5967), 2002.

[4] RJ Bolton and DJ Hand. Unsupervised profiling methods for fraud detection. *Conference on credit scoring and credit control*, 2001.

[5] R.J. Bolton and D.J. Hand. Statistical fraud detection: A review. *Statistical Science*, 17(3):235–249, 2002.

[6] L Breiman. Random forests. *Machine learning*, pages 5–32, 2001.

[7] NV Chawla, KW Bowyer, L.O. Hall, and W.P. Kegelmeyer. SMOTE: synthetic minority oversampling technique. *Journal of Artificial Intelligence Research2*, 16:321–357, 2.

[8] Zengan Gao and Mao Ye. A framework for data mining-based anti-money laundering research. *Journal of Money Laundering Control*, 10(2):170–179, 2007.

[9] John Hunt. The new frontier of money laundering: how terrorist organizations use cyberlaundering to fund their activities, and how governments are trying to stop them. *Information & Communications Technology Law*, 20(2):133–152, June 2011.

[10] L I U Junqiang. Optimal Anonymization for Transaction. *Chinese Journal of Electronics*, 20(2), 2011.

[11] P.J. Lin, B. Samadi, Alan Cipolone, D.R. Jeske, Sean Cox, C. Rendon, Douglas Holt, and Rui Xiao. Development of a synthetic data set generator for building and testing information discovery systems. In *Information Technology: New Generations, 2006. ITNG 2006. Third International Conference on*, pages 707–712. IEEE, 2006.

[12] C M Macal and M J North. Tutorial on agent-based modelling and simulation. *Journal of Simulation*, 4(3):151–162, September 2010.

[13] Dan Magnusson. The costs of implementing the anti-money laundering regulations in Sweden. *Journal of Money Laundering Control*, 12(2):101–112, 2009.

[14] J Pavon, M Arroyo, S Hassan, and C Sansores. Agent-based modelling and simulation for the analysis of social patterns. *Pattern Recognition Letters*, 29(8):1039–1048, June 2008.

[15] Clifton Phua, Vincent Lee, Kate Smith, and Ross Gayler. A comprehensive survey of data mining-based fraud detection research. *Arxiv preprint arXiv:1009.6119*, 2010.

[16] J.R. Quinlan. Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234, 1987.

[17] Agus Sudjianto, Sheela Nair, Ming Yuan, Aijun Zhang, Daniel Kern, and Fernando Cela-Díaz. Statistical Methods for Fighting Financial Crimes. *Technometrics*, 52(1):5–19, February 2010.

[18] Yabo Xu, Ke Wang, Ada Wai-chee Fu, Hong Kong, and Philip S Yu. Anonymizing Transaction Databases for Publication. *International Journal*, pages 767–775, 2008.

[19] Dianmin Yue, Xiaodan Wu, Yunfeng Wang, Yue Li, and Chao-Hsien Chu. A Review of Data Mining-Based Financial Fraud Detection Research. In *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, pages 5514–5517. Ieee, September 2007.

[20] ZM Zhang and JJ Salerno. Applying data mining in investigating money laundering crimes. *discovery and data mining*, (Mlc):747, 2003.

The 27th annual workshop of the Swedish Artificial Intelligence Society (SAIS),
14–15 May 2012, Örebro, Sweden
Published by Linköping University Electronic Press
http://www.ep.liu.se/ecp_home/index.en.aspx?issue=071

Interleaving Configuration Planning and Action Planning in Robotic Ecologies

Lia Susana d.C. Silva-Lopez, Lars Karlsson
{lia.silva,lars.karlsson}@aass.oru.se
School of Science and Technology
Örebro University

Abstract

In the context robotic ecologies, the Configuration Planning Problem (CPP) is concerned with ways to create a flow of information and a flow of causality between the members of an ecology and their environment, in such a way that a goal is satisfied. Ways to solve this type of CPP have been devised and we discuss some in this paper. However, most consider only the causal or information aspects of the members of an ecology, or rely on abstract actions and task hierarchies. A problem with considering only causal or information aspects, is that it complicates modelling situations when both occur and relate to each other. A problem with abstract actions and task hierarchies is that it can lead to incompleteness, and require effort and skill to write down. The goal of this paper is to discuss a way to interleave configuration planning with action planning, in which direct interconnections of either causal or information links, are used to solve the problem of building configurations of networked robotic systems. We show with examples how this approach works, discuss some future directions on where we want this work to get and more specifically in the context of the Giraff+ system.

1 Introduction

Robotic ecologies, as proposed by Saffioti and Broxvall in [10], are collections of devices and programs with cognition and communication capabilities, in which the notion of robot emerges from the interaction between the elements of the ecology. In robotic ecologies, the purpose is to use a synergy of small devices to fulfil goals in a flexible way, instead of building complex universal robots. In the Configuration Planning Problem (CPP) for the context of robotic ecologies (See section 2 of this paper), the purpose is to satisfy a goal by interconnecting a set of available components with constraints between them. A configuration here, can be seen as a collection of functional elements that contribute with their capabilities to fulfil a goal, connected to each other by causal links and information channels.

In this paper, we present some preliminary explorations of how to integrate two kinds of interactions between entities in a robot ecology. We consider information flows, for instance from a light sensor to a process that tests whether a certain threshold of light has been exceeded to an actuator for adjusting the blinds of a window. We also consider causal flows, e.g. in order to obtain information using a video camera, the light first has to be switched on. An important difference between these two types of interactions is that information flows can never interfere with each other, whereas causal flows can interfere by changing the state of the environment. For instance, assume that the system also has the goal to keep the room dark because a person is sleeping there. Switching on the lights in order to use the video camera would interfere with that goal. In addition, for the purpose of this paper we consider causality to apply from one time step to a later one, whereas information flows between entities that are active at the same time step. Our work is motivated by the fact that in the current literature for solving the CPP, information inputs and outputs are either treated as preconditions and postconditions [3] or are not considered in the action planning. Moreover, goals are defined as actions to perform, but there is no mechanism for defining information as a goal.

This work is a step towards building a system capable of generating configurations for multiple goals set by an activity recognition system, of dynamically estimating preferences according to different criteria, and of updating state variables as the configurations executes, in a real world scenario. Our system is intended to be a part of the **Giraff+ system** [1], an adaptive system consisting of a Giraff robot, a sensor network, an activity recognition system [7](also called context recognition in Figure 1), and a configuration planner. The purpose of Giraff+ is to help improving quality of independence of the elderly, by providing personalised services of long-term monitoring in their homes and easier social interaction. Examples of services for improving quality of independence, that Giraff+ can provide in a home would be:

- Displaying time-line information of events of interest in combination with other contextual information, such as blood pressure readings after starting the use of a new medication or blood sugar measurements during days where the patient exercises.
- Reminding and assisting the person in important events, such as taking medication or taking blood sugar or heartbeat measurements.
- Providing long term statistics on e.g. amounts of physical activity or sleep.

Giraff+ is going to be tested in more than a dozen real homes, therefore it should be suitable for real application domains.

There are a number of ways in which a person can interact with its surroundings, and a number of ways to associate sensed data to the activities of a person. The role of the activity recognition system is to infer which activities is the human performing, based on information obtained from the sensor network. For more on activity recognition, see Ullberg et al [12], in which some approaches for recognizing human activities are discussed, and an algorithm for enabling long term and continuous activity recognition based on a temporal reasoner is proposed. For enabling activity recognition, raw sensor readings may not be enough to get enough information. There will be cases where the information may not be available directly, and an interaction with the environment may be needed to get it. The role of the configuration planner is to configure the network, according to the changing nature of the environment and the needs of the activity recognition system, in order to observe or change the value of a state variable.

To illustrate the Giraff+ system we will use Figure 1, where an environment is abstracted as state variables

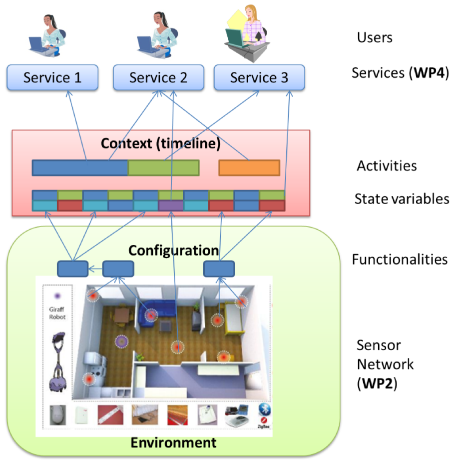

Figure 1: Functional Overview of the Giraff+ System.

that have a value at each point of time. These state variables may represent, for instance, aspects of the layout of the apartment (rooms, furniture), position and motion of individuals in the apartment (e.g. person1 in bedroom, on bed, not moving) or of items in the apartment. State variables may be observable through sensors. But in some cases, to observe a state variable, other state variables need to be changed e.g. to obtain the heart beat rate of a person, the lights of the room where the person is have to be turned on. Also, transformations on raw sensor data may be needed to obtain the value of a state variable e.g. the raw RSSI values between some bluetooth devices needs be transformed into the position of a person, but they can also be transformed into the trajectory of a movement, or transformed into a command for opening a door. Now, a functionality is a program that either operates directly on a sensor or actuator, or processes information from and/or delivers information to other functionalities. Note that a sensor or actuator can perform several functionalities, depending on which programs make use of it. The configuration planner builds configurations that make the sensor network provide the information (state variable values) needed by the activity recognition system, and may also change state variables.

This paper is organised as follows: Section 2 reviews related work, Section 3 explains the conceptual framework for our approach on configuration and action planning, Section 4 presents explanatory examples, and Section 5 describes ideas for future work.

2 Related Work

The CPP and similar problems have been studied in a number of fields, from which Robotic Ecologies and Web Service Composition can be pointed out.

Robotic ecologies, and more particularly ecologies of networked Physically Embedded Intelligent Systems (PEIS) as proposed by Saffioti and Broxvall [10], study how the overall functionalities of robotic systems go beyond the functionality of isolated robots, by adding communication and cognition capabilities into the robots. The CPP in robotic ecologies studies ways to interconnect the members of an ecology, in order to reach a goal.

Lundh et al in [6], propose an approach to automatically generate configurations for a given task in a given environment. An action planner defines which actions need to be accomplished to get to the goal, and for each action, a configuration is made. Methods similar to Hierarchical Task Networks (HTNs) planning were used to hold information on how each action should decompose into configurations. Terminating functionalities were used to determine when a configuration completed its task. Goals were defined in terms of actions. The capabilities of the members of an ecology were modelled as programs that can interact with the environment through sensing or actuating, and/or use information to produce additional information; such programs are called functionalities. Functionalities have causal preconditions, causal postconditions, information inputs and outputs, and also have costs and resources. Preconditions specify under which circumstances the functionality can be used, postconditions indicate how the functionality transforms the world state after its execution, information inputs indicate which information is required for the functionality to execute, and information outputs indicate which information is produced during functionality execution. A different approach is suggested by Gritti in [2], were a reactive configuration algorithm reconfigures the ecology with the available components, in the event of a failure. This approach is appropriate for very dynamic scenarios, because it inherently removes from the search space those functionalities that are not available in the time when they are needed.

Web service composition is concerned with ways of interconnecting self-contained, self-describing, modular applications that can be published, located, and invoked across the web. A composite service is a set of services and the control and data flow among them [3]. A number of approaches have been explored, from genetic algorithms, to neo-classical planning. For example, Tang et al [11] propose a solution to the optimal web service selection problem; this problem deals with selecting web services so that the composite web service give the best overall performance, and it can be seen as the problem of finding an optimal solution to the CPP under similar parameters. The authors propose a genetic algorithm with mutation and knowledge-based crossover as operators, in which each individual is a plan for Web Service Selection; a local optimizer improves individuals in the population, while the crossover operator evaluates constraints between services. In contrast, Peer [8] proposes an approach in which the problem of web service composition is automatically converted into an AI planning problem (see [9]), represented in *PDDL*; after that, the goal and domain description are matched against available AI planners, in order to solve the task.

Our approach to configuration planning is closer to the field of robotic ecologies, and builds on the one of Lundh et al, but unlike them we don't It does not rely on abstract actions or task hierarchies. Instead, we make direct connections between the different components needed to build the configuration, depending on the type of links needed. We do this because, while task hierarchies can speed up planning [9], they can also lead to incompleteness, and require effort and skill to write down, especially every time we add more functionalities into the system.

3 States and Functionalities

The ideas proposed in this paper are built upon the theoretical background presented in [5]. In this way, for the context of this paper, a configuration is a set of elements called functionalities, and the connections between them. Every time a functionality has an effect that changes properties of the world, the state of the world is changed.

3.1 State

A state s holds information that describes the world, in the form of a possible assignment of values to state variables. This assignments can be properties of certain objects (e.g. position(light1) = kitchen, near(light1,stove) = true, near(light1,fridge) = true) or the current state of the different devices on our system (e.g. light1 = on, phone-bt = off).

Time is not explicitly contemplated as part of a state.

3.2 Functionalities

A functionality is a program that interacts with other programs by exchanging data, and/or with the physical world by effecting or observing the value of various properties of entities. These properties are described as state variables. Functionalities can operate on devices, sense, actuate, and may use information to generate information. They can execute simple actions and generate simple information pieces, or execute more complicated tasks and generate more information. For example, we can have a functionality that provides the location of an object as an output and requires no inputs, or we can have many functionalities together in a configuration and give the same output. In this work, we will use the definition by Lundh [5], in Equation 1.

$$f = \langle Id, I, O, \phi, Pr, Po, Re, \mathrm{Cost} \rangle \qquad (1)$$

Every functionality has an Id, a set of inputs I and a set of outputs O that can be of different types, causal preconditions Pr that indicate which values of state variables should hold before the execution of the functionality, causal postconditions Po that state which values of state variables will hold after the execution of the functionality, a transfer function ϕ that represents a transformation from inputs to outputs of the information before execution, to the information after execution, and some resources Re and costs $Cost$. At this early stage of our work, we have no particular definition for costs and resources, but they will receive a great deal of attention later on.

To refer to the element Id of a functionality f, we will write Id_f, and in a similar way we will write Pr_f,Po_f,I_f, and O_f.

4 Connections and Configurations

A connection between functionalities can be formed either by an information link, or by a causal link. Connections are formed when one functionality satisfies an information input or a causal precondition with an information output or a causal postcondition. Configurations are formed by building connections between functionalities. Figure 2 shows an example of a configuration.

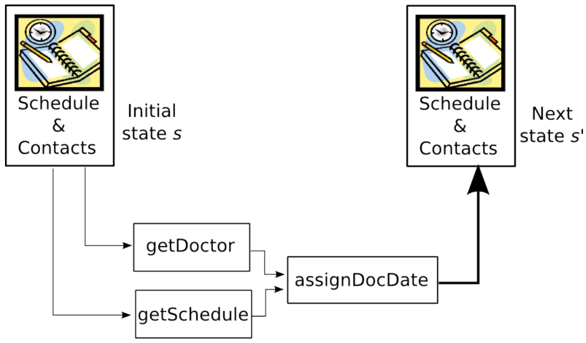

Figure 2: Example of a Configuration. Thin lines are information links. Thick lines are causal links.

4.1 Connections

A connection states a relationship between two functionalities, that can either be defined in terms of an information link, or a causal link.

Information links state that the output k of a functionality a (O_a^k) satisfy the input j of functionality b (I_b^j). Causal links state that the effects k of functionality a (Po_a^k) into the world, satisfy condition j that functionality b requires for execution (Pr_b^j).

The type of link also states the relative execution time of the functionalities in each connection: when two functionalities are related by an information link, they execute within the same interval of time (which we will call time-step), but when they are related by a causal link, the functionality that provides the postcondition for the link executes in a time-step which precedes the one of the functionality that requires such condition. Please note that with time-step we mean a finite interval of time, not a single time point.

In this work, we will use the definition of connections in Equations 2 and 3.

$$Cxi = \langle destF, sourceF, \mathrm{infotype}, rT \rangle \qquad (2)$$

$$Cxc = \langle destF, sourceF, Sv, SvV, rTd, rTs \rangle \qquad (3)$$

Every connection contains a destination functionality *destF*, a source functionality *sourceF*, and a relative execution time rT. The relative execution time in the connection, is used to represent orderings between execution times of the functionalities. Information links contain the type of information required *infotype*, in which the name of the state variable can be used to describe the piece of information that relates the functionalities in the link. Causal links contain a state variable assignment, where *Sv* is the name and *SvV* is the value of the state variable.

For example, suppose we have a causal goal for the system, of setting a slot of our free time with a Doctor's appointment; we have our schedule and our contact list in the form of state variables, functionalities $Id_f = getDoctor$, $Id_f = getSchedule$, and $Id_f = assignDocDate$. Functionality $Id_f = getDoctor$ knows which number is associated to the Doctor and has it as an output. Functionality $Id_f = getSchedule$, gets which slots are available in our schedule and has them as an output. Functionality $Id_f = assignDocDate$, negotiates with the Doctor's appointment system and, as a postcondition, it sets a slot of our free time with the value of a Doctor's, given a number associated to our Doctor and our available times as inputs.

In the case of building a configuration for the given goal by using this elements, we will first have a causal connection between $Id_f = assignDocDate$ as *sourceF*, and the goal as *destF*. In this connection, which *Sv* and *SvV* is a condition of setting a field of the schedule with the value of a doctors appointment, and the execution time of *SourceF*, happens before executing *destF*. In the special case of a goal as a destination functionality, the goal is artificially treated as another functionality; this means that to have the effect we want in the world, this functionality would need to be executed first. We may also have to perform several information connections, for example, connecting $Id_f = getDoctor$ to $Id_f = assignDocDate$ and $Id_f = getSchedule$ to $Id_f = assignDocDate$; in this case, *infotype* would be the piece of information provided in each connection, which can be $phone-nr(Doctor)$ for $Id_f = getDoctor$ to $Id_f = assignDocDate$ and $schedule(freeSpaces)$ for $Id_f = getSchedule$ to $Id_f = assignDocDate$, having all of them a simultaneous execution, since $Id_f = assignDocDate$ needed both pieces of information to execute.

4.2 Configurations

A configuration is a set of connections between a subset F of all functionalities in the world $\boldsymbol{F}$, that satisfy a Goal. In this work, we will use the definition of configuration in Equation 4.

$$c = \langle F, ICX, CCX \rangle \tag{4}$$

Here, F is the set of functionalities used in the configuration, *ICX* is the set of information links, and *CCX* is the set of causal links in the configuration. Relative execution times of functionalities are contained inside each connection.

4.2.1 Admissibility

For a configuration to be admissible, it must not violate constraints on costs, resources, information links and causal links. A proper way to evaluate admissibility on costs and resources will be devised later on in our work, but for the approach presented in this paper, we checked for information admissibility and causal admissibility.

To satisfy information admissibility in a configuration, all inputs in each functionality instance should be connected to the output of another functionality. To satisfy causal admissibility, all Preconditions in a functionality should be satisfied before the execution of the functionality, and no Postconditions of the functionalities in the configuration should be in conflict with the causal links of the configuration. The set of links should be acyclic.

When both information and causal admissibility are satisfied in a configuration, we can say that such configuration is admissible, and consider it a candidate configuration for the goal.

4.2.2 Goals

Since the definition of functionality implies both transforming information and interacting with the physical world, then is reasonable for the goal of a configuration to reflect this. We defined two types of goals: information goals and action goals, in Equations 5 and 6.

$$g_i = \langle Stv \rangle \tag{5}$$

$$g_a = \langle Stv, StvV \rangle \tag{6}$$

Goals are defined in terms of state variables. Stv is a state variable. If the goal is to obtain (or sense) the value of Stv, then it is an information goal. For information goals, the outputs of functionalities are examined to find which one delivers the value of Stv. If the goal is to interact with the world in such a way that we set the value of state variable Stv to $StvV$, then it is an action goal. For action goals, the postconditions of functionalities are examined to find an effect in which the value of Stv is set to $StvV$.

For the planning in the algorithms in Section 5, the Goal can be seen as a Goal Functionality, in which the inputs of the functionality are the information goals, and the preconditions are the action goals.

4.3 Configuration Planning Problem

In our Configuration Planning Problem, given a goal g_i or a goal g_a, a set of available functionalities $\boldsymbol{F}$ and a set of state variable assignments in the start state s, find an *admissible* configuration c such that $F_c \subseteq \boldsymbol{F}$, in such a way that the goal is satisfied.

5 An Algorithm for Configuration Planning

Our first approach to generating configurations with both direct causal and information links between the functionalities, is presented in Algorithms 1 and 2. We have an implementation of this approach in C++, available for downloading in [4].

Algorithm 1 (*ConAc*) requires the assignments of state variables in the start state s, a list of all functionalities in the world $\mathbf{F}$, and a goal g, and delivers a list L of all admissible configurations. L is a global variable for algorithms 1 and 2. *ConAc* builds a functionality $f_g \in \mathbf{F}$, in which the inputs of the functionality are the information goals of g, and the preconditions are the action goals of g. Functionality f_g is added to (currently empty) partial configuration c, and into the list of functionalities of c, F_c, and an initial state functionality (f_i) is used to emulate the conditions of s in order to be able to establish causal links to it, but no causal or information connections have been performed so far, so ICX_c and CCX_c are empty in this assignment. Then, *ConAc* calls *genConf* with the goal functionality as the only member of the partial configuration. Then, *genConf* recursively aggregates all admissible configurations into L. In the end, *ConAc* returns the list L of all admissible configurations suitable for goal g.

Algorithm 2 (*genConf*), requires a partial configuration c, a list of all functionalities in the world $\mathbf{F}$, and the relative time of the last source functionality added into the configuration RT. *genConf* modifies the list of all admissible configurations L by adding admissible configurations as it searches partial configurations. Please note that after execution of Algorithm Algorithm 1, $\mathbf{F}$ contains f_g and f_i.

Algorithm 1 ConAc. State variables in the start state s, available functionalities $\mathbf{F}$, goal g, list of all admissible functionalities L

Require: s,$\mathbf{F}$,g
Ensure: L
build goal functionality $f_g \in \mathbf{F}$ using goal g
build initial functionality $f_i \in \mathbf{F}$ using s
$c \leftarrow \langle \{f_g, f_i\}, \emptyset, \emptyset \rangle$
$L \leftarrow \emptyset$
$genConf(c,\mathbf{F},0)$ */ Algorithm 2 */
if $L == \emptyset$ **then**
    return *failure*
end if
return L

genConf first attempts to satisfy information admissibility by checking on functionalities already existing in the configuration, whose relative execution time does not come after the functionality that we are trying to satisfy. Then, it attempts to satisfy information admissibility with the rest of the functionalities in $\mathbf{F}$. Functionalities connected with information links are executed during the same time step.

If information admissibility is satisfied in a configuration, it stars checking for causal admissibility. *genConf* first checks if causal admissibility can satisfied by f_i. If it can be satisfied by f_i, then causal links are created. If there are still causal links to be satisfied, we choose one *targetcond* in *target* $\in F_c$, executing in timestep *targetime*, and we check if any functionality in the current configuration has a postcondition that can satisfy *targetcond*, as long as its execution time comes before the *targetime*.

On every case where a new connection is created, $genConf$ calls itself recursively with the partial configuration c'.

6 Explanatory Example: Locating the Human

With this example, we intend to show two points. First, we want to show that it is possible to build admissible configurations just by interleaving the right information and causal links between functionalities.

This will be done without the need of HTNs, and without defining terminating functionalities. Second, we want to show that it is possible to define the goal

Algorithm 2 *genConf*. Partial configuration c, available functionalities $\mathbf{F}$

Require: c,$\mathbf{F}$
if inf. admiss. not satisfied **then**
  choose $target \in F_c$ with unsat. in. $targetIn$, executing in $targettime$
  for all $f \in F_c : targetIn \in O_f \wedge f_{executiontime} == targettime$ **do**
    $c' \leftarrow c$
    $append(ICX_{c'}, link(target, f, targetIn, targettime))$
    $genConf(c', \mathbf{F})$
  end for
  for all $f \in (\mathbf{F} \setminus F_c) : targetIn \in O_f$ **do**
    $c' \leftarrow c$
    $append(ICX_{c'}, link(target, f, targetIn, targettime))$
    $append(F_{c'}, f)$
    $genConf(c', \mathbf{F})$
  end for
else if causal admiss. not satisfied **then**
  choose $target \in F_c$ with unsat. prec. $targetcond$, executing in $targettime$
  for all $f \in F_c : targetcond \in Po_f \wedge f_{executiontime} \prec targettime$ **do**
    $c' \leftarrow c$
    $append(CCX_{c'}, link(target, f, targetcond, targettime, f_{executiontime}))$
    $genConf(c', \mathbf{F})$
  end for
  for all $f \in (\mathbf{F} \setminus F_c) : targetcond \in Po_f \wedge noConflict(Po_f, CCX_c)$ **do**
    $c' \leftarrow c$
    $append(CCX_{c'}, link(target, f, target, targettime, targettime + 1))$
    $append(F_{c'}, f)$
    $genConf(c', \mathbf{F})$
  end for
else
  $append(L, c)$
  return 0
end if

of a configuration not only as an action that needs to be performed, but also as information that we want to get from the world.

The state variables and functionalities available in the world are described in Tables 1 and 2. Our world can be described as the home of a person that needs some assistance in the daily life, with matters like localizing medications or health devices, tracking activities, or daily chores.

In this example, an information goal is accomplished with a configuration that contains direct information and causal links. The goal is to provide a position of the cell phone, which we bind to the position of the human. In the state variables, the position of the phone is unknown or outdated, so we need to acquire it again.

Table 1: State Variables.

Name	Value
bt (phone)	off
position (phone)	-
number (phone)	777

To get the position of the phone, Table 2 shows two functionalities that can help, which are $get-ph-GPS$ and $bt-tracker$. Functionality $get-ph-GPS$ obtains the position of the phone as GPS coordinates, if it has as an input the number of the phone. Functionality $bt-tracker$ obtains the position of the phone in terms of how close the phone is to a certain bluetooth source, if it has as an input the number of the phone, and if the

Table 2: Available Functionalities.

Id	Ins	Outs	Prec	Post
get-ph-nr	-	number (phone)	-	
get-ph-GPS	number (phone)	position (phone)	-	
bt-tracker	number (phone)	position (phone)	bt (phone) = on	
turn-phbt-on	number (phone)			bt (phone) = on

bluetooth of the phone is on. In the case of generating a configuration that uses $bt - tracker$, the phone bluetooth is off (Table 1). In order to satisfy causal admissibility, a causal link needs to be used. Such link can be done to functionality $turn - phbt - on$. This functionality turns on the bluetooth in a phone whose number is given as an input; $turn - phbt - on$ can be implemented by combining a program that runs in a local computer and sends an SMS with a code to the desired phone, and another program that when receiving the SMS with the code, turns on the bluetooth in the phone. In each case, the planner will call itself recursively until satisfying all admissibility criteria, reaching a limit size, or after recursively finding that there is no way to satisfy admissibility in any partial configuration, and then a failure is returned. For more details, see Section 5. In figure 3, the configurations obtained by applying our approach in this situation are summarized.

Execution monitoring is not part of this planner, but it if we had a system that monitored when a failure was found on applying a certain functionality, or a functionality is no longer available, then another program could update the description of functionalities, and our planner could be launched again with that feedback. However, a better approach for execution monitoring can be devised later on.

7 Conclusions and Future Work

In this paper, we have presented an algorithm for configuration planning in which functionalities can connect to each other directly by using either action or information links, without the help of a hierarchy definition, just by matching the information and causal needs of every functionality that is added into the configuration. Our planner can handle both causal goals and information goals. The combination of this features allow the execution of an action in order to fulfil an information requirement, and the capture of an information piece in order to execute an action.

Our current approach has a number of limitations that we would like to overcome. However, we would like to keep the idea of direct information and causal links in a configuration and we would like keep the idea of defining goals as interactions with the world or as information pieces to obtain. But we would like a more expressive way of defining goals, and of defining how information and causal requirements relate with the execution time of the functionality; an example of the latter is being able to discern if a functionality changes a state variable or gives an information output either during its execution time, after execution, or from the start of its execution. Also, the planning in the causal dimension in our proposed algorithm, is based on causal links planning without any guidance of heuristics. In order for our planner to scale up to more causally challenging problems, this needs to be addressed.

For the application of our work in the **Giraff+** Project, we want to handle limited resources, multiple goals with preferences, and interactions with activity recognition. More particularly, we are interested on describing preferences for reliability, quality, performance, use of resources and handling of undesired consequences. We are interested on studying interactions between multiple requests, requests that can not be fulfilled because of a conflict or resource scarcity, and on handling disjunctive requests. For challenges associated to real application domains, we are interested on seamless ways of adding and removing functionalities in the system, on handling failures and degradation of performance in configurations, and on execution monitoring.

In order to address many of the previous matters, for the near future we are defining a language for expressing goal preferences, as well as interactions of time with causal and information requirements in functionality definitions. We are also defining heuristics and will use semiring soft constraints for guiding the construction of configurations in a smarter way.

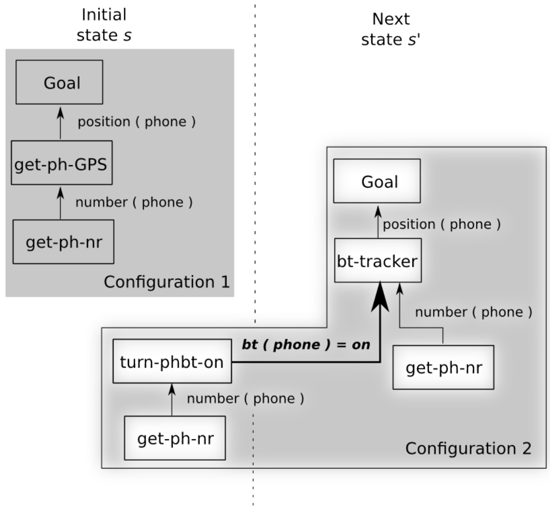

Figure 3: Configurations on Example 'Locating the Human'

Acknowledgements Giraff+ is funded by the European Community's Framework Programme Seven (FP7) under contract #28817

References

[1] Giraff+ website. http://www.giraffplus.eu/.

[2] M. Gritti, M. Broxvall, and A. Saffiotti. Reactive self-configuration of an ecology of robots. In *Proceedings of the ICRA-07 Workshop on Network Robot Systems.*, 2007.

[3] X. Su. J. Rao. A survey of automated web service composition methods. *Semantic Web Services and Web Process Composition. Lecture Notes in Computer Science, Springer*, 3387:43–54, 2005.

[4] LSilvaWebsite. http://aass.oru.se/ ˜lcsz/.

[5] R. Lundh. *Robots that Help Each Other: Self-Configuration of Distributed Robot Systems*. PhD thesis, Orebro University, 2009.

[6] R. Lundh, L. Karlsson, and A. Saffiotti. Autonomous functional configuration of a network robot system. *Robotics and Autonomous Systems, Elsevier*, 56(10):819–830, 2008.

[7] F. Pecora, M. Cirillo, F. Dell'Osa, J. Ullberg, and A. Saffiotti. A constraint-based approach for proactive, context-aware human support. *Journal of Ambient Intelligence and Smart Environments*, 2012. (To appear).

[8] J. Peer. A PDDL based tool for automatic web service composition. *Principles and Practice of Semantic Web Reasoning, Lecture Notes in Computer Science, Springer*, 3208:149–164, 2004.

[9] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, third edition, 2010.

[10] A. Saffiotti and M. Broxvall. Peis ecologies: Ambient intelligence meets autonomous robotics. In *Proc. of the Int. Conf. on Smart Objects and Ambient Intelligence (sOc-EUSAI)*, pages 275–280, 2005.

[11] M. Tang. A hybrid genetic algorithm for the optimal constrained web service selection problem in web service composition. In *Proceeding of the 2010 IEEE World Congress on Computational Intelligence*, 2010.

[12] J. Ullberg, A. Loutfi, and F. Pecora. Towards continuous activity monitoring with temporal constraints. In *Proceedings of the Workshop on Planning and Plan Execution for Real-World Systems at ICAPS09*, 2009.